Kernel Korner - Unionfs: Bringing Filesystems Together

Dec 01, 2004 By Charles P. Wright (/user/1001388) and Erez Zadok (/user/801167)

in

Unionfs merges several directories into a single unified view. We describe applications of Unionfs and also interesting implementation aspects.

(/issue/128)

From Issue #128 December 2004 (/issue/128)

Now home directories from both /alcid and /penguin are available from /home.

Unionfs supports multiple read-write branches, so the user's files will not migrate from one directory to another. This contrasts with previous unioning systems, such as BSD-4.4's Union Mounts, which generally supported only a single read-write branch.

Merging Distribution ISO Images

Most Linux distributions are available as both ISO images and also as individual packages. ISO images are convenient because they can be burnt directly to CD-ROMs, and you need to download and store only a few files. To install to a machine over a network, however, you often need to have the individual packages in a single directory. Using the loopback device, the ISO images can be mounted on separate directories, but this layout is not suitable for a network install because all files need to be located in a single tree. For this reason, many sites maintain copies of both the ISO images and also the individual package files, wasting both disk space and bandwidth and increasing management efforts. Unionfs can alleviate this problem by virtually combining the individual package directories from the ISO images.

In this example, we are mounting over two directories, /mnt/disc1 and /mnt/disc2. The mount command is as follows:

```
# mount -t unionfs -o dirs=/mnt/disc1,/mnt/disc2 \
> none /mnt/merged-distribution
```

Now /mnt/merged-distribution can be exported using NFS or FTP for use in network installs.

Copy-on-Write Unions

In the previous example of the ISO images, all of the branches in the union were read-only; hence, the union itself was read-only. Unionfs also can mix read-only and read-write branches. In this case, the union as a whole is read-write, and Unionfs uses copy-on-write semantics to give the illusion that you can modify files and directories on read-only branches. This could be used to patch a CD-ROM. If the CD-ROM is mounted on /mnt/cdrom, and an empty directory is created in /tmp/cdpatch, then Unionfs can be mounted as follows:

```
# mount -t unionfs -o dirs=/tmp/cdpatch,/mnt/cdrom \
> none /mnt/patched-cdrom
```

When viewed through /mnt/patched-cdrom, it appears as though you can write to the CD-ROM, but all writes actually will take place in /tmp/cdpatch. Writing to read-only branches results in an operation called a copyup. When a read-only file is opened for writing, the file is copied over to a higher-priority branch. If required, Unionfs automatically creates any needed parent directory hierarchy.

In this CD-ROM example, the lower-level filesystem enforces the read-only permissions, and Unionfs respects them. In other situations, the lower-level filesystem may indeed be read-write, but Unionfs should not modify the branch. For example, you may have one branch that contains pristine kernel sources and then use a separate branch for your local changes. Through Unionfs, the pristine sources should be read-only, as the CD-ROM was in the previous example. This can be accomplished by adding =ro to a directory in the dirs mount option. Assume that /home/cpw/linux is empty, and /usr/src/linux contains a Linux kernel source tree. The following mount command makes Unionfs behave as a source code versioning system:

```
# mount -t unionfs -o \
> dirs=/home/cpw/linux:/usr/src/linux=ro \
> none /home/cpw/linux-src
```

This example makes it appear as if an entire Linux source tree exists in /home/cpw/linux-src, but any changes to that source tree, such as changed source files or new object files, actually go to /home/cpw/linux.

With a simple modification, we also can use an overlay mount. That is, we can replace /home/cpw/ linux with the unified view:

```
# mount -t unionfs -o
> dirs=/home/cpw/linux:/usr/src/linux=ro
> none /home/cpw/linux
```

Implementation

Most filesystem operations in Unionfs move from higher-priority branches to lower-priority branches. For example, LOOKUP begins in the highest-priority branch in which the parent exists and then moves to lower-priority branches. During the lookup operation, Unionfs caches information for use in later operations.

CREATE attempts to create files in the highest-priority branch where the parent directory exists. The CREATE operation uses the cached lookup information to operate directly on the appropriate branch, so in effect, it moves from higher-priority branches to lower-priority branches.

Unionfs uses several techniques to provide the illusion of modifying read-only branches, and at the same time, maintains normal UNIX semantics. If there is an error while creating a file, error handling must be performed. Error handling proceeds from lower-priority branches to higher-priority branches. Starting in the highest-priority branch in which the parent exists, Unionfs attempts to create the file in each higher-priority branch. Finally, if the operation fails in the highest-priority branch for the whole union, then Unionfs returns an error to the user.

In contrast to CREATE, the UNLINK operation always proceeds from lower-priority branches to higher-priority branches. Because the last underlying object to be UNLINKed is the highest-priority object, the user-visible state is not modified until the very end of Unionfs's UNLINK operation. The most complex situations to handle are partial errors. If removing an intermediate file fails and Unionfs simply removes the highest-priority file, a lower-priority file becomes visible to the user. To handle these error conditions, Unionfs uses a special high-priority file called a whiteout. If Unionfs encounters a whiteout, it behaves as if the file does not exist in any lower-priority branch. Internally, to create a whiteout for a file named F, Unionfs creates a zero-length file named .wh.F. Getting back to UNLINK—if an intermediate UNLINK has failed, instead of deleting the highest-priority file, Unionfs renames the file to the corresponding whiteout name.

This careful ordering of operations has two effects. The first is that UNIX semantics are maintained even in the face of errors or read-only branches. The user-visible state isn't modified until the operation is attempted on the highest-priority branch. The success or failure of the operation is determined by the success or failure of this branch. Through the use of whiteouts, a file can be deleted even if it exists on a read-only branch. The second effect is that when no errors occur, files and directories tend to stay in the branches where they were originally. This is important because one of the goals of Unionfs is to keep the files in separate places.

Comments

Comment viewing options

Threaded list - expanded ▼ | Date - newest first ▼ | 50 comments per page ▼ | Save settings

Select your preferred way to display the comments and click "Save settings" to activate your changes.

"I think UnionFS is great (/article/7714#comment-324324)
Submitted by Thinking (not verified) on Tue, 07/29/2008 - 14:35.

"I think UnionFS is great for Live CD applications(e.g. DSL/Linux - many others). Also, in general UnionFS is great for small simple applications like remembering writes (COWs - copy on writes) to any read-only file system (e.g. CDROM). However, beyond that, and especially the tomato example I think UnionFS is actually far too dangerous for serious computing. Why? Because there is no intelligent way to remember/decide the duplicate file issues. That being said I'm actually very pro-UnionFS; but like seasoning a fine stew - a little seasoning goes a long way, too much seasoning spoils the stew - re: a little UnionFS is great, beyond that it is DANGEROUS".
adipex phentermine (http://forums.usfirst.org/member.php?u=12324)
buy cheap phentermine online (http://www.forum.psoft.net/member.php?u=21124)
buy cheap phentermine (http://www.mountwashington.org/forums/member.php?u=10194)
buy phentermine online (http://www.forum.psoft.net/member.php?u=21125)

UnionFS - great for some apps, very dangerous for others (/article/7714#comment-302554)
Submitted by mjc777 (http://www.embeddedrelated.com/blogs-1/nf/Mark_Conrad.php) (not verified) on Sat, 10/27/2007 - 13:04.

I think UnionFS is great for Live CD applications(e.g. DSL/Linux - many others). Also, in general UnionFS is great for small simple applications like remembering writes (COWs - copy on writes) to any read-only file system (e.g. CDROM). However, beyond that, and especially the tomato example I think UnionFS is actually far too dangerous for serious computing. Why? Because there is no intelligent way to remember/decide the duplicate file issues. That being said I'm actually very pro-UnionFS; but like seasoning a fine stew - a little seasoning goes a long way, too much seasoning spoils the stew - re: a little UnionFS is great, beyond that it is DANGEROUS (IMHO).

unionfs snapshots of live mounted systems (/article/7714#comment-137773)
Submitted by Cesar (not verified) on Thu, 05/04/2006 - 21:50.

Is it possible to create snapshots of live mounted systems using unionfs? If so how to do it? I would like some help on this issue. I currently use LVM2 snapshots to do consistent backups of live running systems and are having some problems: system hangs, hard locks, etc. If I could do the same as with LVM2 snapshots but with the unionfs it would be great.
thanks

ISO-File from unionfs (/article/7714#comment-55057)
Submitted by pmt (not verified) on Wed, 06/08/2005 - 07:56.

Is it possible to make a iso-file from unionfs?

I have tried it with mkisofs, but the directory depth is too high.
What i can do?

re:ISO-File from unionfs (/article/7714#comment-128427)
Submitted by Jake Miller (http://www.portalano.net) (not verified) on Tue, 01/31/2006 - 03:53.

Hi Joost,
Is it the unionfs stuff discussed recently here used for slax?

Unknown:
A little more involved to modify the image, perhaps check the slax site for hints. Note: .iso images are built with mkisofs, not written to like a read/write filesystem. This _is_ documented, seek ;)

Cheers,

re:ISO-File from unionfs (/article/7714#comment-128429)
Submitted by Randall M. (http://www.tuttipagina.org) (not verified) on Tue, 01/31/2006 - 03:56.

Yes, NetBSD 3.0 has support for compressed VND images - they work exactly
the same way as e.g. in Knoppix on Linux or Freesbie on FreeBSD. The kernel needs to have support enabled via the VND_COMPRESSION kernel
option (see options(4)). After that, you can create a disk image using vnd(4), and then compress that image with vndcompress(1), put it on the CD, configure it with vnconfig(8)'s -z option, then mount as usual.
You can use UFS or CD9660 as filesystem, your choice, vnd(4) won't care.

re:ISO-File from unionfs (/article/7714#comment-128428)
Submitted by Bill Manson (http://www.tutticompletti.com) (not verified) on Tue, 01/31/2006 - 03:55.

apparently, unionfs is a standard part of slax. but of course it is only operating when you're booting from the slax cd, not if you're mounting it
as an iso. and even then, it only creates the illusion of a modifiable fs, the cd itself isn't changed.

Re:ISO-File from unionfs (/article/7714#comment-127103)
Submitted by Anonymous (not verified) on Tue, 12/06/2005 - 17:34.

Do it with ex2fs . An Image using ex2fs (or other native Linux fs) can be mounted/read ro in Linux without a problem

error in code examples? (/article/7714#comment-27543)
Submitted by Anonymous (not verified) on Sun, 04/03/2005 - 10:37.

In some code examples there is a "," and in some a ":" between the two "dirs=". I get an error when using ",", with ":" it works.

NFS exports (/article/7714#comment-14669)
Submitted by Anonymous (not verified) on Mon, 01/03/2005 - 22:16.

Most problems reported above (compile-time error, Unknown error 524 while doing ls, etc) are fixed in v1.0.4.

However, although the article claims that it is possible to NFS-export and mount a unionfs FS, I get a "getfh failed: Operation not permitted" from mountd when I try to mount it from remote. If I export a parent directory of the unionfs dir, I can mount the parent but cd-ing to the unionfs dir shows an empty directory. Has anyone ever suceeded in NFS-exporting and remote mounting a unionfs directory?

Cheers,
Gabor

Re: NFS exports (/article/7714#comment-26431)
Submitted by Geoff Mishkin (http://amsa.dnsalias.org) (not verified) on Tue, 03/15/2005 - 18:47.

I have a similar problem in version 1.0.9. When I try to mount the unionized directory, I get:

```
mount: amsa.dnsalias.org:/mnt/amsa-e failed, reason given by server: Permission denied
```

However, in the server logs, it would seem that the mount is successful:

```
Mar 15 18:21:12 [rpc.mountd] authenticated mount request from comm700-c0501-dhcp088.bu.edu:707
for /mnt/amsa-e (/mnt/amsa-e)
```

However, the packet trace indicates a return status of `ERR_ACCESS` to the mount request.

Anyone else have a problem like this?

The official NEWS file in the (/article/7714#comment-38175)
Submitted by Anonymous (not verified) on Thu, 05/05/2005 - 13:14.

The official NEWS file in the current download (1.0.11) admits that NFS was broken in 1.0.9 on 2.6 kernels, and claims this to be fixed in 1.0.11.

NFS Exports (/article/7714#comment-27869)
Submitted by Ian Rawlings (not verified) on Mon, 04/04/2005 - 15:26.

Yep, also having exactly the same problem, it appears to be impossible to export a unionfs via NFS. Just to test I wasn't doing anything wrong I have tried exporting the parent directory of a unionfs mountpoint and was able to see the directory and the unionfs mountpoint, but trying to list the contents of the unionfs mountpoint (i.e. the contents of the unionfs itself) lead to a frozen command that eventually freed up with a "permission denied" once I'd restarted the NFS server.

A great shame, UnionFS would be very useful for me and my discless clients, but it appears that it's not compatible with NFS. Note that the initial "authenticated mount request" in the logs means that mountd was happy with the mount request, so it's the NFS daemons/kernel threads that are refusing to co-operate.

RE: NFS (/article/7714#comment-65751)
Submitted by derFlo (not verified) on Wed, 06/15/2005 - 23:45.

official mailinglist:
http://www.fsl.cs.sunysb.edu/pipermail/unionfs/2005-June/000610.html
(http://www.fsl.cs.sunysb.edu/pipermail/unionfs/2005-June/000610.html)
cite:knfsd will not export Unionfs unless you
have an fsid option in /etc/exports

http://www.fsl.cs.sunysb.edu/pipermail/unionfs/2005-April/000350.html
(http://www.fsl.cs.sunysb.edu/pipermail/unionfs/2005-April/000350.html)
cite:/mnt/unionfs *(rw,fsid=42,async)

i dont know what this fsid means (goin to sleep now), but just (touch) created a file throu a nfs mounted directory that showed up in the rw branch of my union.. (debian kernel-image-2.6-11-1k7 , module-assistent,unionfs unionfs-source 1.0.11-1)

unionfs for 2.6 (/article/7714#comment-14611)
Submitted by Anonymous (not verified) on Wed, 12/29/2004 - 08:01.

Is unionFS is available for 2.6 kernels?

Article and related home page lack download link (/article/7714#comment-13464)
Submitted by Anonymous (not verified) on Sun, 11/07/2004 - 02:53.

I was quite interested in the unionfs article in the newest issue but can not find any link to download the unionfs software, even from the project's homepage. Please talk with the author to resolve this oversight.

Software available here (/article/7714#comment-13546)
Submitted by Anonymous (not verified) on Mon, 11/22/2004 - 10:19.

ftp://ftp.fsl.cs.sunysb.edu/pub/unionfs (ftp://ftp.fsl.cs.sunysb.edu/pub/unionfs)

Though, I've been trying to get it to compile with 2.4.27/28 and had no luck so far...

fix f. kernel 2.4.27 (/article/7714#comment-14848)
Submitted by Anonymous (not verified) on Wed, 01/12/2005 - 22:18.

if it's about an error concerning implicit declaration of IS_IMMUTABLE, try this:

```
--- namei.c.org Thu Jan 13 03:18:06 2005
+++ namei.c Thu Jan 13 03:38:17 2005
@@ -67,7 +67,7 @@
if (IS_APPEND(dir))
return -EPERM;
if (check_sticky(dir, victim->d_inode)||IS_APPEND(victim->d_inode)||
- IS_IMMUTABLE(victim->d_inode))
+ IS_IMMUTABLE_FILE(victim->d_inode)||IS_IMMUTABLE_LINK(victim->d_inode))
return -EPERM;
if (isdir) {
if (!S_ISDIR(victim->d_inode->i_mode))
```

make CC=gcc-3.0

http://www.fsl.cs.sunysb.edu/ (/article/7714#comment-13547)
Submitted by Anonymous (not verified) on Mon, 11/22/2004 - 10:27.

http://www.fsl.cs.sunysb.edu/project-unionfs.html
(http://www.fsl.cs.sunysb.edu/project-unionfs.html)

Oops. This is the author's page for the above link. Sorry

/src/unionfs-1.0.3# make all (/article/7714#comment-13548)
Submitted by Anonymous (not verified) on Mon, 11/22/2004 - 10:58.

/src/unionfs-1.0.3# make all
gcc -D__KERNEL__ -DMODULE -DFISTGEN -I. -I/lib/modules/`uname -r`/build/include -Wall -Wno-unused-label -Werror -g -O2 -c -o subr.o subr.c
In file included from /lib/modules/2.4.27-1-k7/build/include/linux/spinlock.h:6,
from /lib/modules/2.4.27-1-k7/build/include/linux/wait.h:16,
from /lib/modules/2.4.27-1-k7/build/include/linux/fs.h:12,
from /lib/modules/2.4.27-1-k7/build/include/linux/capability.h:17,
from /lib/modules/2.4.27-1-k7/build/include/linux/binfmts.h:5,
from /lib/modules/2.4.27-1-k7/build/include/linux/sched.h:9,
from fist.h:32,
from subr.c:22:
/lib/modules/2.4.27-1-k7/build/include/asm/system.h: In function `__set_64bit_var':
/lib/modules/2.4.27-1-k7/build/include/asm/system.h:190: warning: dereferencing type-punned pointer will break strict-aliasing rules
/lib/modules/2.4.27-1-k7/build/include/asm/system.h:190: warning:

dereferencing type-punned pointer will break strict-aliasing rules
make: *** [subr.o] Error 1

I get the same results with a virgin 2.4.28 source code too.

Anyone?

Got unionfs working (/article/7714#comment-13592)
Submitted by Steve (not verified) on Sat, 11/27/2004 - 18:25.

I finally got unionfs working after...

Downgrading from 2.6.9 to 2.4.28
and
Downgrading from gcc 3.3.4 to gcc 3.2.3 (just for the build)

I hope this helps.

Steve

Spoke too soon... (/article/7714#comment-13597)
Submitted by Anonymous (not verified) on Sun, 11/28/2004 - 12:15.

I've tried searching google, but with no luck. First of I discovered that the command to mount a unionfs is mount -t unionfs -o dir=/dirone:/dirtwo none /combined ... you'll note the : as the article uses a , separator. The other problem I'm having is running 'ls -l' on the /combined directory give me multiple "Unknown error 524" for all subdirectoried, thought it seems to be working.

Perfect! It works here too. J (/article/7714#comment-13596)
Submitted by Anonymous (not verified) on Sun, 11/28/2004 - 04:12.

Perfect! It works here too. Just incase anyone is interested. I'm using Debian Sarge and just installed the gcc-3.2 compiler. Then changed the symlink gcc to point to gcc-3.2.

Thanks Steve! :-)