ZACK'S KERNEL NEWS

New Networked Filesystem POHMELFS

Evgeniy Polyakov announced POHMELFS (Parallel Optimized Host Message Exchange Layered File System), a new high-performance networked filesystem. As Evgeniy explained, the protocol used by POHMELFS, although not yet entirely stable, appears to be superior to that of NFS in every test conducted. Also, it is a user-space filesystem that can work on top of any lower level filesystem (ext2, etc).

Andi Kleen asked for an explanation of how reliable the filesystem is, and Evgeniy indicated that it was still a bit early for people to use. Evgeniy was able to do large filesystem operations and generate identical checksums on the client and server. He said that he expected to find bugs, and he also expected to find areas of POSIX non-compliance. He said, "it works for [the] usual NFS-like workload, but since it is [a] rather young FS (this particular design implementation exists for about [a] couple of months), there may be some questions … ."

The Linux kernel mailing list comprises the core of Linux development activities. Traffic volumes are immense, often reaching ten thousand messages in a given week, and keeping up to date with the entire scope of development is a virtually impossible task for one person. One of the few brave souls to take on this task is Zack Brown. Our regular monthly column keeps you abreast of the latest discussions and decisions, selected and summarized by Zack. Zack has been publishing a weekly online digest, the Kernel Traffic newsletter for over five years now. Even reading Kernel Traffic alone can be a time consuming task. Linux Magazine now provides you with the quintessence of Linux Kernel activities, straight from the horse's mouth.

Maintainership Changes

Following Hans Reiser's murder conviction, the ReiserFS developers, led by Edward Shishkin, have migrated ReiserFS development off of their Namesys systems and onto kernel.org and have updated the MAINTAINERS file to point to the new locations. Because the current ResierFS developers have previously been dependent on Namesys for salaries that have kept them working on the project, it's unclear what will happen to them. My guess is that they will be joined or gradually replaced by new volunteers who are interested in helping out and interested in the code itself. ReiserFS is still a fairly groundbreaking filesystem, with a lot to offer.

Greg Kroah-Hartman announced that Jesse Barnes would be taking over as the PCI maintainer, and they orchestrated an orderly transfer of git repositories.

Timur Tabi posted a patch to the MAINTAINERS file, listing himself as the official maintainer of the Cirrus Logic CS4270 sound driver, the Freescale QUICC engine library, the QUICC engine UCC UART driver, and the Freescale SoC sound drivers.

Zhang Wei posted a patch removing himself as the official maintainer of the Freescale DMA driver, and listing Li Yang in his place. Bob Copeland posted a patch to add the OMFS filesystem to the MAINTAINERS file and listed himself as the official maintainer.

Avi Kivity posted a patch to add a maintainer entry for KVM (Kernel-based Virtual Machine) on PowerPC and listed Hollis Blanchard as the official maintainer. Avi also created an entry for KVM on the S390, listing Carsten Otte and Christian Borntraeger as the official co-maintainers.

New Transmitter Driver

Martin Kebert posted a new driver for the Zhen Hua PPM-4CH RC transmitter, which, among other things, is found in little toy helicopters from Walkera. Although there was no discussion, it looks like the sort of patch that would be accepted quickly.

New Flash Filesystem

Artem Bityutskiy and Adrian Hunter announced the creation of a new filesystem for Flash drives, as a joint effort between Nokia and the University of Szeged. The filesystem is called UBIFS, and according to the developers, it is sleek, fast, and nearly ready for widespread use. Their tests showed that UBIFS has better speed and scalability than JFFS2 (from which they borrowed many ideas).

Jan Engelhardt also asked how UBIFS compared with LogFS, and Adrian said they were less familiar with LogFS. According to the small amount of testing they'd done, LogFS appeared slower and had fewer features, and they couldn't tell how well it would scale.

LogFS also has a smaller code base, which is good, and is not dependent on the UBI module, which is also good, according to Jörn Engel.

Artem pointed out that LogFS had been discussed at the 2005 Linux Kongress and is still not ready for widespread use, which is why he said it was pointless to compare the two. In fact, he said he "refused" to compare the two, which ended up ruffling some feathers.

Pekka Enberg pointed out that folks might be curious about why the UBIFS developers weren't content to wait for LogFS to improve sufficiently and that this was why a comparison between the two systems would be significant.

As Jörn put it, there was no reason not to merge both filesystems, and he'd been planning to put out a new LogFS patch soon.

Meanwhile, Artem fanned his own flames by asking Jörn to list "all crucial features which are not implemented (but have to) like bad eraseblock handling when you send the patch-set?"

The only interesting thing to come out of the discussion was that it does seem true that UBIFS is nearly fully usable and loaded with features. Also, as Artem affirmed, it does not scale well to large Flash devices (e.g., 64GB). For that, he says, the UBI module would have to be reworked to some degree.

Reinventing the Lock

Someone is always trying to fix kernel locking. Originally, it was the Big Kernel Lock, then that was split into a bunch of smaller locks. Now a range of locks is used throughout the kernel. Recently, Matthew Wilcox wanted to replace semaphores with mutexes, spinlocks, and completions wherever possible. In some areas it wouldn't be easy, so Matthew also wanted to identify all semaphores and clarify what they should be replaced with and what workarounds to add for cases that had no direct translations available.

Daniel Walker was also into this idea and already had been submitting his own patches bit by bit. He pointed out that converting mutexes to spinlocks wouldn't really be necessary, unless a serious performance issue was at stake. Christoph Hellwig said spinlocks used less memory and were even completely optimized out of the kernel in single-processor systems, whereas mutexes were larger and stayed in the kernel whether or not they were needed.

David Chinner asked Matthew what was going to happen regarding semaphores that were harder to convert to anything else, and Matthew explained that a bunch of different types of semaphores are in question, each with their own special qualities. One of the features that distinguishes semaphores from other kinds of locks is that semaphores don't just lock up a resource – they manage resource locking for a number of available resource items. Other kinds of locks are more like on/off switches, without regard to how many of a given thing might be available for exclusive use at any given time.

To replace the variety of semaphores in use in the kernel, Matthew, Arjan van de Ven, and Ingo Molnár have been designing something they call *kcounter*, a mechanism for replacing the semaphore resource-counting feature with a cookie that can be picked up and put down again by anything making use of those resources. Matthew says this wouldn't precisely mimic semaphore behavior, but it would bridge part of the gap between semaphores and more binary-oriented locking techniques.

David was not pleased with the cookie-based solution, which had been tried in the kernel in other areas, resulting in what he called "an ugly, ugly API." He had no alternative to suggest in this case, so he and Matthew started discussing the specifics of how the existing semaphores behaved and how that might be replaced with some other lock plus *kcounter*. The thread ended with no specific solutions to the harder cases, but it's clear that kernel locking should be improving soon, which will result in faster multi-processor systems and, hopefully, if unneeded locks can be optimized away, slightly faster single-processor systems as well.

Linux Magazine Exclusive