

Re: Future Linux on Bistable Storage

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2008-06/msg00735.html>

- *From:* Daniel Barkalow <barkalow@xxxxxxxxxxxx>
 - *Date:* Mon, 2 Jun 2008 15:30:19 -0400 (EDT)
-

On Mon, 2 Jun 2008, Rick van Rein wrote:

Hello,

Future generations of Linux are likely to run on machines with non-volatile memories based on bistable technologies. This will save the energy of DRAM refresh cycles and avoid the mechanical problems related to hard disks. The result is probably a computer with no distinction between disk and RAM.

One major difference between disk and RAM is the tradeoffs between size, speed, and price. It's highly unlikely that any one technology will ever beat all others in both of the usage patterns which are normal for RAM and disk in devices considered by their users to be "computers".

Such computer architectures can conserve a lot of energy, as it is very simple to suspend and resume them: no data loss means no time needed to resume the system -- except perhaps for I/O initialisation. Conserving energy means (1) having more fun/hours per kg of batteries, and (2) saving the planet.

I wonder in how far Linux is (already) capable of dealing with such hardware. Several core concepts suddenly become meaningless if disk and RAM are merged into one storage device:

- * swapping out data

CONFIG_SWAP=n

- * buffering disk blocks

I think current ramdisk code skips it.

Re: Future Linux on Bistable Storage

- * lazy loading of programs from disk to RAM
- * mapping and unmapping disk onto RAM

You still need the distinction between what's in a process's address space and what's in the filesystem, in order to have a system that feels like a computer.

Regarding I/O init: Current Linux boots while initiating one device at a time. Some degree of serialism must exist, but after scanning a bus would it not be possible to fork threads to initiate each of the devices found on it? Am I overlooking details or could this be a useful change?

Userspace setup stuff hugely dominates boot time. Some drivers, and it's not known which, only successfully initialize on the assumption that some other drivers are not initializing hardware at the same time. This has been tried, and it (a) had some hard-to-track bugs and (b) didn't help enough to be worth dealing with.

If it is possible, it would make Linux *_much_* quicker to boot (also on current hardware), and it could even make it acceptable to do a suspend after a *_very_* short time (say, a minute) of non-activity.

Writing all of RAM to disk burns more power than powering RAM for several minutes, even assuming you don't have to spin up the hard drive. For that matter, restarting the system will power up all of the devices on the bus that you've shut down because you're not using them.

It'd be an easy way to save some trees :)

You have a wood-fired power plant?

In the long run, it seems fair to assume that all devices, including processors, would use non-volatile buffers based on one of the bistable semiconductor devices currently being developed; but I would not be surprised if at first we will be working with computers that have older parts mixed in; boards that still forget their registers when switched off.

In general, any time you switch drivers or kernel versions, you'll want to have the device return all of its registers to a default state and take it through an initialization process. For that matter, even if the memory keeps its state, on power failure some calculations will have been done and some will not, so the resulting storage will be in some unknown state and

Re: Future Linux on Bistable Storage

need to be reinitialized from storage that wasn't changing.

But in the end I expect the entire system to switch itself off in between every two keystrokes (!) simply because loosing no data anywhere in the system means being able to resume operations at the next whim/interrupt.

It's likely that the power cost of getting a consistant state to the bistable memory will be more than the idle power required for a couple of seconds.

–Daniel

This .sig left intentionally blabk

—

To unsubscribe from this list: send the line "unsubscribe linux-kernel" in the body of a message to majordomo@xxxxxxxxxxxxxxxx

More majordomo info at <http://vger.kernel.org/majordomo-info.html>

Please read the FAQ at <http://www.tux.org/lkml/>