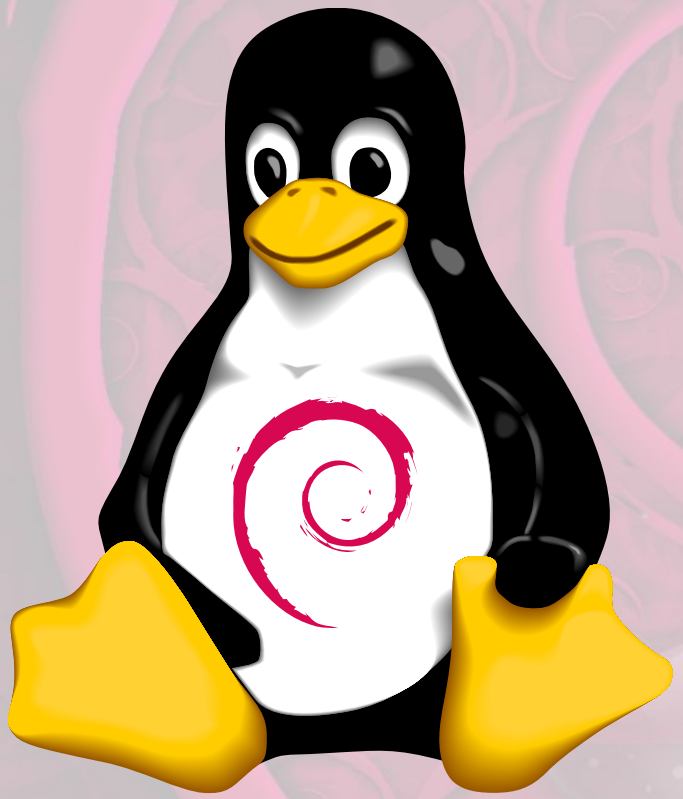
What's new in the Linux kernel

and what's missing in Debian



Ben Hutchings

DebConf 15

Ben Hutchings

- Professional software engineer by day, Debian developer by night (or sometimes the other way round)
- Regular Linux contributor in both roles since 2008
- Working on various drivers and kernel code in my day job
- Debian kernel and LTS team member, now doing most of the kernel maintenance aside from ports
- Maintaining Linux 3.2.y stable update series on kernel.org

Linux releases early and often

- Linux is released about 5 times a year (plus stable updates every week or two)
  - ...though some features aren't ready to use when they first appear in a release
- Since my talk last year, Linus has made 5 releases (3.17-4.1), and 4.2 is on the verge of release
- Good news: we have lots of new kernel features in testing/unstable
- Bad news: some of them won't really work without new userland

Recap of last year's features

- Lustre userland support is still missing – currently blocked on a small licence issue
- bedup was deprecated in favour of duperemove – which still needs a sponsor (#784898)
- liblockdep was not packaged, so I spent some hours on it and it's now in binary-NEW
- arm64 and ppc64el were included in jessie including useful kernel packages
- NVMe, SCSI disk and virtio block drivers support the block multiqueue interface

Extended BPF [3.17..] (1)

- Berkeley Packet Filter (BPF) is a BSD kernel facility to accelerate tcpdump by running packet filter code in kernel
- Filter code is interpreted in a VM, but can save a lot of copying so it's a net win
- Linux implements a compatible VM and can also use it for syscall filtering (seccomp mode 2), firewalling (xt_bpf) and network scheduling (act_bpf, cls_bpf)
- Higher packet rates and new applications make BPF performance more important
- JIT compilation implemented for many architectures (arm, arm64 mips, powerpc, sparc64, x86) starting in 3.0, but disabled by default
- VM is 32-bit with only 2 registers, so doesn't make good use of modern CPU capabilities even with JIT

Extended BPF [3.17..] (2)

- Extended BPF (eBPF) better suited to modern CPUs and applications:
  - Conditional branches have only one destination
  - 10 64-bit registers
  - Instructions for byte order conversion, arithmetic right shift, atomic add, ...
  - Associative arrays (hash-maps) shared with userland
- Usable for packet filtering, network scheduling and kprobe tracepoint filtering
- JIT compilation implemented for arm64 and x86_64
- BPF interpreter replaced by eBPF interpreter and converter, improves performance even with JIT disabled
- Coming soon: compile (restricted) C to eBPF using clang

overlayfs [3.18]

- A new(ish) union file-system
- Simpler than aufs, resulting in some limitations:
  - Doesn't work on top of remote file-systems such as NFS (yet)
  - Can't be exported via NFS
  - White-outs require an inode each
  - Fills in holes when copying-up sparse files
  - Only supports one writeable branch
  - Creating a hard link requires copy-up

switchdev [3.19]

- Linux is widely used on network appliances with integrated switches – configured using vendor-specific APIs
- Many PCIe network cards also include switches for use with virtualisation (macvlan or SR-IOV) – configured using netlink API
- Linux also includes software bridge driver (slow) – configured using different netlink API, or ioctls
- New 'switchdev' concept provides common driver interface for configuring all of these
  - Supported by i40e, ixgbe, qlcnic, rocker, macvlan
  - Each port is a net device; use ethtool etc. to configure link
  - 'bridge' command from iproute configures static L2 forwarding rules
- L2 learning and L3 forwarding can be offloaded or done in software depending on hardware capabilities

Atomic mode-setting [ongoing] (1)

- Kernel Mode-Setting (KMS) removed need for X video drivers to configure display hardware directly
- Video display generator has one or more pipelines (“CRTCs”)
- Each pipeline takes input from one or more frame-buffers (“planes”) - background, cursor, video, ...
- Each pipeline's output is routed to one or more screens (“connectors”) through signal encoders
- KMS allows changing the inputs and outputs, changing refresh rate, etc., but not all at once
  - May result in flickering or tearing, or may fail because intermediate state is not supported even though intended final state is

Atomic mode-setting [ongoing] (2)

- Display generator can compose multiple planes using less power than a general GPU
- Window system will need to reconfigure pipeline quite often, so flickering and tearing are undesirable
- Atomic mode-setting API allows setting entire configuration as a transaction – atomically applied or rejected
  - And all changes can be synchronised to vblank
- Needs driver changes to support it
  - Mostly complete for i915 [4.2], msm, tegra drivers
- Needs userland to take advantage of it
  - Changes to Xorg and Wayland are still in development

Live patching [4.0]

- Kernel upgrades require a reboot (or kexec) to complete
  - Disruptive if you haven't embraced cloud computing
  - But often essential to close security holes
- Live patching of the kernel offers a way to fix *some* bugs without a reboot
- First implemented for Linux by Ksplice (now Oracle) – free software but closed development, only for OEL/Fedora/Ubuntu
- RH and SUSE each reimplemented live patching – eventually agreed common code to go upstream
- Would be nice to use this in Debian for stable security updates, but increases work needed for each update
- Anyone want to work on this in the kernel team (or pay a developer)?

NVDIMMs [4.0] (1)

- Flash storage arrays keep getting faster
- New non-volatile memory (NVM) technologies may be faster and more durable
- NVM as fast disk (SATA, NVMe) worked up to a point
- NVM on memory bus (NVDIMM) makes more sense if it's fast enough...but it still shouldn't be rewritten as often as DRAM
- Two possible access modes for NVDIMMs
  - Map NVM to fixed physical memory addresses (PMEM)
  - Provide several memory-mapped apertures to configurable regions in NVM (BLK)
- NVDIMMs may be partitioned into PMEM and BLK regions

NVDIMMs [4.0] (2)

- PMEM mode allows mapping directly into processes without copying (DAX)
  - But if it fails, those processes crash
  - Dependent on file-system support – so far supported by ext2, ext4, xfs [4.2]
- BLK mode allows adding RAID layer and hot-swapping faulty modules
  - But it requires copying to and from DRAM, so is slower

Encryption in ext4 [4.1]

- eCryptfs provides encryption in a layered filesystem; available since Linux 2.6.19
- Why replace general solution with extension to just one filesystem?
  - Performance: avoids double caching
  - Can depend on xattrs and other features not included in all file-systems
  - More flexible – allows choice of which directories to encrypt without capability to mount
- f2fs added same interface [4.2]

Intel MPX [3.19]

- MPX (Memory Protection Extensions) provide efficient array bounds checking without C/C++ ABI changes
- Implemented in the newest Intel processors (codename Skylake)
- Requires changes in kernel, toolchain, libraries to set up and use bounds tables, mostly in unstable:
  - Linux 3.19
  - gcc 5.1
  - binutils 2.25
  - glibc 2.20 [experimental]
- Hardware released this month

Batched network transmit [3.18]

- Network stack calls driver's `ndo_start_xmit` operation to send each skb – one packet or a multi-packet chunk of TCP data
- Drivers could not know when the next packet will be sent, so would have to write a hardware register every time
- When most traffic is not TCP or not sent in large chunks, this means a lot of slow writes to the hardware – limiting packet send rate
  - On virtualised hardware, register writes are even more expensive
- Kernel now sets a flag in skb to indicate whether it will immediately pass more packets
- Drivers can use this flag to decide when they need to write to the hardware
  - Supported by many 10G/40G Ethernet drivers, some 1G Ethernet drivers, hv_netvsc and virtio_net

Y2038 compliance [ongoing]

- Unix APIs use `time_t` to represent time in seconds since the epoch (start of 1970)
- On 32-bit architectures `time_t` is 32-bit (`int` or `long`), so time values will wrap in early 2038
- Embedded Linux systems will be running on 32-bit CPUs for a long time yet... maybe long enough for this to be a problem
- New system calls, ioctls and C library changes needed to support 64-bit `time_t` (`long long`)
  - Will probably be opt-in at compile time, like Large File Support
- Some internal interfaces and drivers are also being fixed to work beyond 2038

Questions?