Kernel upgrades

June 2019
Alexei Starovoitov
ast@kernel.org

We say "I work on the linux kernel". What does it mean?

- We add new features and fixing bugs.
- We want people to run the latest kernels.
  - To find more bugs in our new features ☺
- To get feedback faster about new features.

No extensibility via kernel modules

- Long ago we made a strategic decision to make BPF not extensible via kernel modules.
- Companies contribute BPF helpers/features.
- Companies (datacenters who contributed new BPF feature) want new kernel version.
- Android wants new kernel version.
- Customers are pushing distro vendors to ship newer kernels,
- because they've heard good thing about BPF features.
- All of it is happening because new features require kernel upgrade.

Positive cycle

- We are happy, since our features and bug fixes get to users faster.
- Faster kernel upgrade cycle accelerates kernel development.
- Demand for faster development means more kernel developers.
- More developers -> more diversity -> healthier community -> better decisions.
- To maintain this positive cycle new features must be the core kernel.

No features in kernel modules

- New features must not be in the kernel modules.
- This rule applies to drivers too.
- Think what it will take to generalize your driver feature to be suitable for networking core and push it there. Even if your driver is the only user.
- It's ok for it to be rough. It will get generalized.

kernel/driver interface

- kernel<->driver interface was great 10 years ago.
- napi_gro_receive and ndo_start_xmit are not enough in the era of XDP, AF_XDP, ktls/tc offloads.
- page alloc, skb alloc, xdp frame alloc should be in the core.
- The kernel is struggling to extend this driver interface, since new features are still part of the driver.
- NIC vendors need to have drivers for many kernel versions, hence little incentive to improve core.
- I propose Dave to reject patches that add features to drivers.

Backports are evil

- Backports bring new features to older kernel.
- Users lose an incentive to upgrade.
- It's harder for us (kernel developers) to fix bugs.
- Avoid backports. Upgrade kernels faster.

Scaling kernel development process

June 2019

Alexei Starovoitov
ast@kernel.org

Email is dying.

Email is dead. Long live Gmail.

- 'dynamic email' = integration with G suite. Feels like webpage.
- Delays will get longer.
- Kernel community has to try an alternative.

This alternative can be github.

- How many of you send pull-req on github ?
- How many of you merged pull-req ?

Step 1

- bi-directional sync of kernel.org/bpf-next and github/.../bpf-next
- Developers can choose both mechanisms to send patches.
- Doubles the work for myself and Daniel.
- No auto Acks in github.

Step 2

- github subscribes to bpf@vger and recognizes emails [PATCH bpf-next 0/N] and creates them as PR on github
    - Makes it easy for maintainers to apply patches with single click.
- github sends emails to bpf@vger for PR submitted on github

Step 3

- github recognizes replies with Acked-by and discussion. Injects them into github ui.
- comments on github are sent as emails to bpf@vger

Win-win

- At this point developers and maintainers can send old school emails or use UI

- Further misc steps:
    - merge of PR on github closes corresponding thread in patchworks
    - close PRs on github when patches were pushed manually
    - continuous integration with build bots on github