From `bpf_prog_pack` to new `module_alloc`

Song Liu

Overview

- What: Let BPF programs (and more) share huge pages
- Why:
  - `vmalloc()` + `set_memory_*()` fragments direct map, and thus slows down the system over time
  - Reduce iTLB pressure

v1: `bpf_prog_pack`

- Reserves 2MiB for small bpf programs
- Naive allocator
- In upstream 5.18+ for x86_64; powerpc WIP

v2: `execmem_alloc`

- Reuse `vmap_area` allocator
- Can be used for bpf program, bpf trampoline, module text, ftrace, kprobe, etc.
- Made LWN, but didn’t make through reviews

v3: new `module_alloc`

- Goal: Enable huge pages for modules (text and data), BPF programs, and more

- Part 1: `module_layout` => `module_memory`, landed in 6.4-rc1
- Part 2: Universal `module_alloc()` for all architectures (suggested by Thomas Gleixner, WIP)
- Part 3: Reuse `vmap_area` allocator (similar to v2)
- Part 4: Let bpf program etc. use the new `module_alloc()`

Thanks!

Questions?