# uprobe multi link

jiri olsa / isovalent

# UPROBE

- attach anywhere in user space
- attach on USDT probes
  (User Statically-Defined Tracing)
- uprobe: path/offset/PID
- usdt: +ref_ctr_offset

# UPROBE

- **uprobe is based on perf event**

```
bpf(BPF_PROG_LOAD, {prog_type=BPF_PROG_TYPE_KPROBE, ...) = 10
perf_event_open({type=0x7 /* PERF_TYPE_??? */,     ... ) = 9

   uprobe_register
      sets sw breakpoint on inode:offset

ioctl(9, PERF_EVENT_IOC_SET_BPF, 10)    = 0
ioctl(9, PERF_EVENT_IOC_ENABLE, 0)      = 0
```

## MULTIPLE UPROBES

- attach of many uprobes takes time
- extra file descriptor for each uprobe

```
26.49%  bpftrace  [kernel.vmlinux]        [k] smp_call_function_single
|
---smp_call_function_single
   |
   |--25.92%--perf_install_in_context
   |          __do_sys_perf_event_open
   |          do_syscall_64
   |          entry_SYSCALL_64_after_hwframe
   |          syscall
   |          |
   |           --25.89%--bpf_attach_probe
   |                     bpf_attach_uprobe
   |                     bpftrace::AttachedProbe::attach_uprobe
   |                     bpftrace::AttachedProbe::AttachedProbe
   |                     std::make_unique<bpftrace::AttachedProbe,...
   |                     bpftrace::BPFtrace::attach_probe
   |                     bpftrace::BPFtrace::run
   |                     main
   |                     __libc_start_call_main
   |
    --0.57%--event_function_call
```

```
# BPFTRACE_MAX_PROBES=100000  perf stat -e cycles,instructions \
  ./bpftrace -e 'uprobe:./uprobe_multi:uprobe_multi_func_* { } \
                   i:ms:1 { exit(); }'

Attaching 1001 probes...

      19.279262539 seconds time elapsed
```

```
/proc/51548/fd/0 -> /dev/pts/2
/proc/51548/fd/1 -> 'pipe:[94709]'
/proc/51548/fd/2 -> 'pipe:[94709]'
/proc/51548/fd/3 -> anon_inode:bpf-map
/proc/51548/fd/4 -> 'anon_inode:[eventpoll]'
/proc/51548/fd/5 -> 'anon_inode:[perf_event]'
/proc/51548/fd/6 -> 'anon_inode:[perf_event]'
/proc/51548/fd/7 -> 'anon_inode:[perf_event]'
/proc/51548/fd/8 -> 'anon_inode:[perf_event]'

/proc/51548/fd/9 -> 'anon_inode:[perf_event]'   UPROBE
/proc/51548/fd/10 -> anon_inode:bpf-prog

/proc/51548/fd/11 -> 'anon_inode:[perf_event]'
/proc/51548/fd/12 -> anon_inode:bpf-prog

/proc/51548/fd/13 -> 'anon_inode:[perf_event]'
/proc/51548/fd/14 -> anon_inode:bpf-prog

/proc/51548/fd/15 -> 'anon_inode:[perf_event]'
/proc/51548/fd/16 -> anon_inode:bpf-prog

/proc/51548/fd/17 -> 'anon_inode:[perf_event]'
/proc/51548/fd/18 -> anon_inode:bpf-prog
...
```

# UPROBE MULTI LINK

- **attach bpf program to multiple uprobes**

  **https://lore.kernel.org/bpf/20230424160447.2005755-1-jolsa@kernel.org/**

```
bpf(BPF_PROG_LOAD, {prog_type=BPF_PROG_TYPE_KPROBE, ...) = 10
bpf(BPF_LINK_CREATE, {prog_fd = 10, ...

   for each uprobe {
     uprobe_register
       sets sw breakpoint on inode:offset
   }

) = 11
```

# UPROBE MULTI LINK

- **new bpf_attr API**

```
struct { /* struct used by BPF_LINK_CREATE command */

  struct {
          __u32           flags;
          __u32           cnt;
          __aligned_u64   paths;
          __aligned_u64   offsets;
          __aligned_u64   ref_ctr_offsets;
          __aligned_u64   cookies;
  } uprobe_multi;

} link_create;
```

**uprobe-X: paths[X]-offsets[X]-refctr[X]-cookies[X]**

# LIBBPF API

```
int bpf_link_create(int prog_fd, int target_fd,
    enum bpf_attach_type attach_type,
    const struct bpf_link_create_opts *opts);

struct bpf_link_create_opts {
        union {
                ...
                struct {
                        __u32 flags;
                        __u32 cnt;
                        const char **paths;
                        const unsigned long *offsets;
                        const unsigned long *ref_ctr_offsets;
                        const __u64 *cookies;
                } uprobe_multi;
        };
        size_t :0;
};
```

# LIBBPF API

```
struct bpf_link*
bpf_program__attach_uprobe_multi_opts(
    const struct bpf_program *prog,
    const char *binary_path,
    const char *func_pattern,
    const struct bpf_uprobe_multi_opts *opts);

struct bpf_uprobe_multi_opts {
        size_t sz;
        const char **paths;
        const char **syms;
        const unsigned long *offsets;
        const unsigned long *ref_ctr_offsets;
        const __u64 *cookies;
        size_t cnt;
        bool retprobe;
        size_t :0;
};
```

# LIBBPF API

- **libbpf sections**

SEC("**uprobe.multi**/path:function_pattern")

SEC("**uprobe.multi.s**/path:function_pattern")

```
SEC("uprobe.multi/./uprobe_multi:uprobe_multi_func_*")
int test_uprobe_bench(struct pt_regs *ctx)
{
       return 0;
}
```

- **use multi link to attach USDTs in**

```
bpf_program__attach_usdt
```

# HELPERS

- bpf_get_func_ip(ctx)
- bpf_get_attach_cookie(ctx)

# PID FILTER

- current uprobe can specify PID filter
- not in uprobe_multi RFC

# thanks, questions..