redhat.

Debugging OVS with GDB (macros)

Eelco Chaudron
Senior Software Engineer
December 2018

How to debug Open vSwitch (vswitchd)

- Use the various ovs-..ctl commands
- Enable *debug* logging
- Custom build if the issue can be replicated
- Attach GDB to the live process

If you get a core dump, GDB is your only option...

Using GDB

The Open vSwitch included script

- OVS has a bundled Python GDB script in `./utilities/gdb/ovs_gdb.py`
- Can be loaded into GDB with the `source` command or add it to your `~/.gdbinit`
- For example:

```
$ gdb $(which ovs-vswitchd) $(pidof ovs-vswitchd)
GNU gdb (GDB) Red Hat Enterprise Linux 7.6.1-110.el7
Copyright (C) 2013 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
...
(gdb) source utilities/gdb/ovs_gdb.py
(gdb) ovs_<tab><tab>
ovs_dump_bridge                 ovs_dump_netdev_provider
ovs_dump_bridge_ports           ovs_dump_ovs_list
ovs_dump_dp_netdev              ovs_dump_simap
ovs_dump_dp_netdev_poll_threads ovs_dump_udpif_keys
ovs_dump_dp_netdev_ports        ovs_show_fdb
ovs_dump_dp_provider            ovs_show_upcall
ovs_dump_netdev
```

Using GDB

Examining data structures

- Manually walking OVS data structures is time consuming
- The GDB Python script has build in support for the following:
  - `ForEachCMAP  struct cmap/cmap_node`
  - `ForEachHMAP  struct hmap/hmap_node`
  - `ForEachLIST  struct ovs_list`
  - `ForEachNL  struct nlattr`
  - `ForEachSHASH  struct shash/shash_node`
  - `ForEachSIMAP  struct simap/simap_node`
  - `ForEachSMAP  struct smap/smap_node`
- The above classes are iterators which allow mapping to a specific data structure

Available GDB commands

- All commands start with **ovs_** so it's easy to identify them
- Use `ovs_<tab><tab>` to show all available commands
  - *ovs_dump_xx* commands dump *raw* information
  - *ovs_show_xx* commands mimimic ovs-..ctl like commands
- Use `help ovs_<cmd>` to shows detailed help

```
(gdb) help ovs_dump_udpif_keys
Dump all nodes of an ovs_list give
    Usage: ovs_dump_udpif_keys {<udpif_name>|<udpif_address>} {short}

      <udpif_name>    : Full name of the udpif's dpif to dump
      <udpif_address> : Address of the udpif structure to dump. If both the
                        <udpif_name> and <udpif_address> are omitted the
                        available udpif structures are displayed.
      short           : Only dump ukey structure addresses, no content details
```

Dump commands

- Can have options to dump additional information
- Dump commands show structure addresses so they can be easily copied

```
(gdb) ovs_dump_bridge ports
(struct bridge *) 0x1dbeb50: name = br-test, type = system
    (struct port *) 0x1da79d0: name = br-test, brige = (struct bridge *) 0x1dbeb50
        (struct iface *) 0x1e009e0: name = 0x1d5b410 "br-test", ofp_port = 65534, netdev = ...
(struct bridge *) 0x1d5d950: name = ovs_pvp_br0, type = netdev
    (struct port *) 0x1d9e880: name = dpdk0, brige = (struct bridge *) 0x1d5d950
        (struct iface *) 0x1da8c90: name = 0x1da3250 "dpdk0", ofp_port = 1, netdev = ...
    (struct port *) 0x1dbe7d0: name = ovs_pvp_br0, brige = (struct bridge *) 0x1d5d950
        (struct iface *) 0x1dbe970: name = 0x1dbdc70 "ovs_pvp_br0", ofp_port = 65534, netdev ...
    (struct port *) 0x1d58e50: name = vhost0, brige = (struct bridge *) 0x1d5d950
        (struct iface *) 0x1dbcd50: name = 0x1d5ac90 "vhost0", ofp_port = 2, netdev = …

(gdb) p *(struct port *) 0x1d9e880
$5 = {
  hmap_node = {
    hash = 3315928014,
    next = 0x0
  },
  ...
```

Show commands

- Show commands try to mimic ovs-..ctl commands
- The optional *dbg* option will include cut/pastable structure addresses

```
(gdb) ovs_show_upcall
netdev@ovs-netdev:
  flows         : (current 0) (avg 0) (max 0) (limit 10000)
  dump duration : 1ms
  ufid enabled  : true

  55: (keys 0)
  59: (keys 0)
  ...

(gdb) ovs_show_upcall dbg
netdev@ovs-netdev, ((struct udpif *) 0x1d603a0):
  flows         : (current 0) (avg 0) (max 0) (limit 10000)
  dump duration : 1ms
  ufid enabled  : true

  55: (keys 0), ((struct revalidator *) 0x1d9ccc0)
  59: (keys 0), ((struct revalidator *) 0x1d9ccd8)
```

Examples

ovs_dump_ovs_list and ovs_dump_simap

- ovs_dump_ovs_list, ovs_dump_simap and ovs_dump_smap are general structure dump routines
- Should be added for all other iterators from slide 4

```
(gdb) ovs_dump_ovs_list &ovsrcu_threads 'struct ovsrcu_perthread' list_node dump
(struct ovsrcu_perthread *) 0x7f2a14000900 =
  {list_node = {prev = 0xf48e80 <ovsrcu_threads>, next = 0x7f2acc000900}, mutex...

(struct ovsrcu_perthread *) 0x7f2acc000900 =
  {list_node = {prev = 0x7f2a14000900, next = 0x7f2a680668d0}, mutex ...

(struct ovsrcu_perthread *) 0x7f2a680668d0 =
  {list_node = {prev = 0x7f2acc000900, next = 0xf48e80 <ovsrcu_threads>}, …
```

```
(gdb) p usage
$5 = (const struct simap *) 0x7ffde4cf9f50
(gdb) ovs_dump_simap 0x7ffde4cf9f50
handlers    : 40 / 0x28
ports       : 5 / 0x5
revalidators: 16 / 0x10
rules       : 10 / 0xa
```

```
Breakpoint 1, trtcm_policer_qos_construct
    (details=0x135bad0, conf=0x7ffd31f5da28)
(gdb) ovs_dump_smap details
cbs: 2048
cir: 151800
eir: 151800
pbs: 2048
```

Examples

Continue

```
(gdb) ovs_dump_bridge
(struct bridge *) 0x1dbeb50: name = br-test, type = system
(struct bridge *) 0x1d5d950: name = ovs_pvp_br0, type = netdev

(gdb) ovs_dump_bridge_ports 0x1dbeb50
(struct port *) 0x1da79d0: name = br-test, brige = (struct bridge *) 0x1dbeb50
    (struct iface *) 0x1e009e0: name = 0x1d5b410 "br-test", ofp_port = 65534, netdev = …

(gdb) ovs_dump_dp_netdev
(struct dp_netdev *) 0x7fe33eb9f010: name = ovs-netdev, class = (struct dpif_class *) 0x935160...

(gdb) ovs_dump_dp_netdev_ports 0x7fe33eb9f010
(struct dp_netdev_port *) 0x1d5fce0:
    port_no = 0, n_rxq = 1, type = tap
    netdev = (struct netdev *) 0x1d5fa20: name = ovs-netdev, n_txq/rxq = 1/1
...

(gdb) ovs_dump_dp_netdev_poll_threads 0x7fe33eb9f010
(struct dp_netdev_pmd_thread *) 0x7fe314731010: core_id = 1, numa_id 0
(struct dp_netdev_pmd_thread *) 0x7fe2eef36010: core_id = 15, numa_id 0
(struct dp_netdev_pmd_thread *) 0x7fe32b737010: core_id = 4294967295, numa_id 2147483647
```

Examples

Continue

```
(gdb) ovs_dump_dp_provider
(struct registered_dpif_class *) 0x1d5c440: (struct dpif_class *) 0x935160 = {type = ...
(struct registered_dpif_class *) 0x1d5d460: (struct dpif_class *) 0x961d80 = {type = …
...

(gdb) ovs_dump_netdev
(struct netdev *) 0x1e00a50: name = vxlan_sys_4789 , auto_classified = false, netdev_class = ...
(struct netdev *) 0x1d5fa20: name = ovs-netdev     , auto_classified = false, netdev_class = ...
(struct netdev *) 0x1e006b0: name = br-test        , auto_classified = false, netdev_class = ...
...

(gdb) ovs_dump_udpif_keys
(struct udpif *) 0x1d603a0: name = netdev@ovs-netdev, total keys = 256

(gdb) ovs_dump_udpif_keys 0x1d603a0
(struct umap *) 0x1d95188:
  (struct udpif_key *) 0x7fe28c051b40: key_len = 140, mask_len = 152
                                       ufid = a49eaa74-94c1-75fe-7e61-4e50fc82263d
                                       hash = 0x77cedc01, pmd_id = 15
                                       state = UKEY_OPERATIONAL
                                       n_packets = 181394, n_bytes = 10883640
                                       used = 354021661, tcp_flags = 0x0000
  (struct udpif_key *) 0x7fe28c042320: key_len = 140, mask_len = 152
                                       ...
```

Examples

Continue

```
(gdb) ovs_show_fdb ovs_pvp_br0 dbg hash
[(struct mac_learning *) 0x1d9cb50]
table.n          : 2
secret           : 0xdcecb1b2
idle_time        : 300
max_entries      : 8192
ref_count        : 2
need_revalidate : false
ports_by_ptr.n   : 2
ports_by_usage.n: 2
total_learned    : 2
total_expired    : 0
total_evicted    : 0
total_moved      : 0

FDB "hash" table:
port                VLAN  MAC                Age out @
-----------------  ----  -----------------  ---------
01[dpdk0]              0  00:00:01:00:00:00     354559 [(struct mac_entry *) 0x7fe28803c4a0]
02[vhost0]             0  00:00:02:00:00:00     354559 [(struct mac_entry *) 0x7fe288006560]

Total MAC entries: 2

Current time is between 354258 and 354263 seconds.
```

Add your own

- Add your own commands to the script, and share…
- Adding a command is easy, for example:

```
class CmdDumpDpProvider(gdb.Command):
    """Dump all registered registered_dpif_class structures.
    Usage: ovs_dump_dp_provider
    """
    def __init__(self):
        super(CmdDumpDpProvider, self).__init__("ovs_dump_dp_provider",
                                                gdb.COMMAND_DATA)

    def invoke(self, arg, from_tty):
        dp_providers = get_global_variable('dpif_classes')
        if dp_providers is None:
            return

        for dp_class in ForEachSHASH(dp_providers,
                                     typeobj="struct registered_dpif_class"):

            print("(struct registered_dpif_class *) {}: "
                  "(struct dpif_class *) 0x{:x} = {{type = {}, ...}}, "
                  "refcount = {}".
                  format(dp_class,
                         long(dp_class['dpif_class']),
                         dp_class['dpif_class']['type'].string(),
                         dp_class['refcount']))
```

redhat.

THANK YOU

plus.google.com/+RedHat

facebook.com/redhatinc

linkedin.com/company/red-hat

twitter.com/RedHat

youtube.com/user/RedHatVideos