NAME

mlx5dv_dr_domain_create, mlx5dv_dr_domain_sync, mlx5dv_dr_domain_destroy – Manage flow domains

mlx5dv_dr_table_create, mlx5dv_dr_table_destroy – Manage flow tables

mlx5dv_dr_matcher_create, mlx5dv_dr_matcher_destroy – Manage flow matchers

mlx5dv_dr_rule_create, mlx5dv_dr_rule_destroy – Manage flow rules

mlx5dv_dr_action_create_drop – Create drop action

mlx5dv_dr_action_create_tag – Create tag actions

mlx5dv_dr_action_create_dest_ibv_qp, mlx5dv_dr_action_create_dest_table, mlx5dv_dr_action_create_dest_vport – Create packet destination actions

mlx5dv_dr_action_create_packet_reformat – Create packet reformat actions

mlx5dv_dr_action_create_modify_header – Create modify header actions

mlx5dv_dr_action_create_flow_counter – Create devx flow counter actions

mlx5dv_dr_action_destroy – Destroy actions

SYNOPSIS

```
#include <infiniband/mlx5dv.h>

struct mlx5dv_dr_domain *mlx5dv_dr_domain_create(
        struct ibv_context *ctx,
        enum mlx5dv_dr_domain_type type);

int mlx5dv_dr_domain_sync(
        struct mlx5dv_dr_domain *domain,
        uint32_t flags);

int mlx5dv_dr_domain_destroy(struct mlx5dv_dr_domain *domain);

struct mlx5dv_dr_table *mlx5dv_dr_table_create(
        struct mlx5dv_dr_domain *domain,
        uint32_t level);

int mlx5dv_dr_table_destroy(struct mlx5dv_dr_table *table);

struct mlx5dv_dr_matcher *mlx5dv_dr_matcher_create(
        struct mlx5dv_dr_table *table,
        uint16_t priority,
        uint8_t match_criteria_enable,
        struct mlx5dv_flow_match_parameters *mask);

int mlx5dv_dr_matcher_destroy(struct mlx5dv_dr_matcher *matcher);

struct mlx5dv_dr_rule *mlx5dv_dr_rule_create(
        struct mlx5dv_dr_matcher *matcher,
        struct mlx5dv_flow_match_parameters *value,
        size_t num_actions,
        struct mlx5dv_dr_action *actions[]);

void mlx5dv_dr_rule_destroy(struct mlx5dv_dr_rule *rule);

struct mlx5dv_dr_action *mlx5dv_dr_action_create_drop(void);
```

```
struct mlx5dv_dr_action *mlx5dv_dr_action_create_tag(
        uint32_t tag_value);

struct mlx5dv_dr_action *mlx5dv_dr_action_create_dest_ibv_qp(
        struct ibv_qp *ibqp);

struct mlx5dv_dr_action *mlx5dv_dr_action_create_dest_table(
        struct mlx5dv_dr_table *table);

struct mlx5dv_dr_action *mlx5dv_dr_action_create_dest_vport(
        struct mlx5dv_dr_domain *domain,
        uint32_t vport);

struct mlx5dv_dr_action *mlx5dv_dr_action_create_packet_reformat(
        struct mlx5dv_dr_domain *domain,
        uint32_t flags,
        enum mlx5dv_flow_action_packet_reformat_type reformat_type,
        size_t data_sz, void *data);

struct mlx5dv_dr_action *mlx5dv_dr_action_create_modify_header(
        struct mlx5dv_dr_domain *domain,
        uint32_t flags,
        size_t actions_sz,
        __be64 actions[]);

struct mlx5dv_dr_action *mlx5dv_dr_action_create_flow_counter(
        struct mlx5dv_devx_obj *devx_obj,
        uint32_t offset);

int mlx5dv_dr_action_destroy(struct mlx5dv_dr_action *action);
```

DESCRIPTION

The Direct Rule API (mlx5dv_dr_*) allows complete access by verbs application to the device‘s packet steering functionality.

Steering flow rules are the combination of attributes with a match pattern and a list of actions. Rules can have several distinct actions (such as counting, encapsulating, decapsulating before redirecting packets to a particular queue or port, etc.). In order to manage the rule execution order for the packet processing matching by HW, multiple flow tables in an ordered chain and multiple flow matchers sorted by priorities are defined.

Domain

mlx5dv_dr_domain_create() creates a DR domain object to be used with *mlx5dv_dr_table_create()* and *mlx5dv_dr_action_create_*()*.

A domain should be destroyed by calling *mlx5dv_dr_domain_destroy()* once all depended resources are released.

The device support the following domains types:

MLX5DV_DR_DOMAIN_TYPE_NIC_RX Manage ethernet packets received on the NIC. Packets in this domain can be dropped, dispatched to QP‘s, modified or redirected to additional tables inside the domain. Default behavior: Drop packet.

MLX5DV_DR_DOMAIN_TYPE_NIC_TX Manage ethernet packets transmit on the NIC. Packets in this domain can be dropped, modified or redirected to additional tables inside the domain. Default behavior: Forward packet to NIC vport (to eSwitch or wire).

MLX5DV_DR_DOMAIN_TYPE_FDB Manage ethernet packets in the eSwitch Forwarding Data Base

for packets received from wire or from any other vport. Packets in this domain can be dropped, dispatched to vport, modified or redirected to additional tables inside the domain. Default behavior: Forward packet to eSwitch manager vport.

mlx5dv_dr_domain_sync() is used in order to flush the rule submission queue. By default, rules in a domain are updated in HW asynchronously. **flags** should be a set of type *enum mlx5dv_dr_domain_sync_flags*:

MLX5DV_DR_DOMAIN_SYNC_FLAGS_SW: block until completion of all software queued tasks.

MLX5DV_DR_DOMAIN_SYNC_FLAGS_HW: clear the steering HW cache to enforce next packet hits the latest rules, in addition to the SW SYNC handling.

Table

mlx5dv_dr_table_create() creates a DR table in the **domain**, at the appropriate **level**, and can be used with *mlx5dv_dr_matcher_create()* and *mlx5dv_dr_action_create_dest_table()*. All packets start traversing the steering domain tree at table **level** zero (0). Using rule and action, packets can by redirected to other tables in the domain.

A table should be destroyed by calling *mlx5dv_dr_table_destroy()* once all depended resources are released.

Matcher

mlx5dv_dr_matcher_create() create a matcher object in **table**, at sorted **priority** (lower value is check first). A matcher can hold multiple rules, all with identical **mask** of type *struct mlx5dv_flow_match_parameters* which represents the exact attributes to be compared by HW steering. The **match_criteria_enable** and **mask** are defined in a device spec format. Only the fields that where masked in the *matcher* should be filled by the rule in *mlx5dv_dr_rule_create()*.

A matcher should be destroyed by calling *mlx5dv_dr_matcher_destroy()* once all depended resources are released.

Actions

A set of action create API are defined by *mlx5dv_dr_action_create_*()*. All action are created as *struct mlx5dv_dr_action*. An action should be destroyed by calling *mlx5dv_dr_action_destroy()* once all depended rules are destroyed.

When an action handle is reused for multiple rules, the same action will be executed. e.g.: action 'count' will count multiple flows rules on the same HW flow counter context. action 'drop' will drop packets of different rule from any matcher.

Action: Drop *mlx5dv_dr_action_create_drop* create a terminating action which drops packets. Can not be mixed with Destination actions.

Action: Tag *mlx5dv_dr_action_create_tag* creates a non−terminating action which tags packets with **tag_value**. The **tag_value** is available in the CQE of the packet received. Valid only on domain type NIC_RX.

Action: Destination *mlx5dv_dr_action_create_dest_ibv_qp* creates a terminating action delivering the packet to a QP, defined by **ibqp**. Valid only on domain type NIC_RX. *mlx5dv_dr_action_create_dest_table* creates a forwarding action to another flow table, defined by **table**. The destination **table** must be from the same domain with a level higher than zero. *mlx5dv_dr_action_create_dest_vport* creates a forwarding action to a **vport** on the same **domain**. Valid only on domain type FDB.

Action: Packet Reformat *mlx5dv_dr_action_create_packet_reformat* create a packet reformat context and action in the **domain**. The **reformat_type**, **data_sz** and **data** are defined in *man mlx5dv_create_flow_action_packet_reformat*.

Action: Modify Header *mlx5dv_dr_action_create_modify_header* create a modify header context and action in the **domain**. The **actions_sz** and **actions** are defined in *man mlx5dv_create_flow_action_modify_header*.

Action: Flow Count *mlx5dv_dr_action_create_flow_counter* creates a flow counter action from a DEVX

flow counter object, based on **devx_obj** and specific counter index from **offset** in the counter bulk.

Action Flags: action **flags** can be set to one of the types of *enum mlx5dv_dr_action_flags*:

MLX5DV_DR_ACTION_FLAGS_ROOT_LEVEL: is used to indicate the action is targeted for flow table in level=0 (ROOT) of the specific domain.

Rule

mlx5dv_dr_rule_create() creates a HW steering rule entry in **matcher**. The **value** of type *struct mlx5dv_flow_match_parameters* holds the exact attribute values of the steering rule to be matched, in a device spec format. Only the fields that where masked in the *matcher* should be filled. HW will perform the set of **num_actions** from the **action** array of type *struct mlx5dv_dr_action*, once a packet matches the exact **value** of the rule (referred to as a 'hit').

mlx5dv_dr_rule_destroy() destroys the rule.

RETURN VALUE

The create API calls will return a pointer to the relevant object: table, matcher, action, rule. on failure, NULL will be returned and errno will be set.

The destroy API calls will returns 0 on success, or the value of errno on failure (which indicates the failure reason).

LIMITATIONS

Application can verify is a feature is supported by *trail and error*. No capabilities are exposed, as the combination of all the options exposed are way to large to define.

Tables are size less by definition. They are expected to grow and shrink to accommodate for all rules, according to driver capabilities. Once reaching a limit, an error is returned.

Matchers in same priority, in the same table, will have undefined ordered.

A rule with identical value pattern to another rule on a given matcher are rejected.

IP version in matcher mask and rule should be equal and set to 4, 6 or 0. # SEE ALSO

mlx5dv_open_device(3), **mlx5dv_create_flow_action_packet_reformat(3)**, **mlx5dv_create_flow_action_modify_header(3)**.

AUTHOR

Alex Rosenbaum <alexr@mellanox.com> Alex Vesker <valex@mellanox.com>