OPENFABRICS ALLIANCE

10TH ANNUAL INTERNATIONAL DEVELOPER WORKSHOP

# On Demand Paging (ODP) Update

Liran Liss
Mellanox Technologies

# Agenda

- Introduction
- Implementation notes
- APIs and usage
- Statistics
- What's new

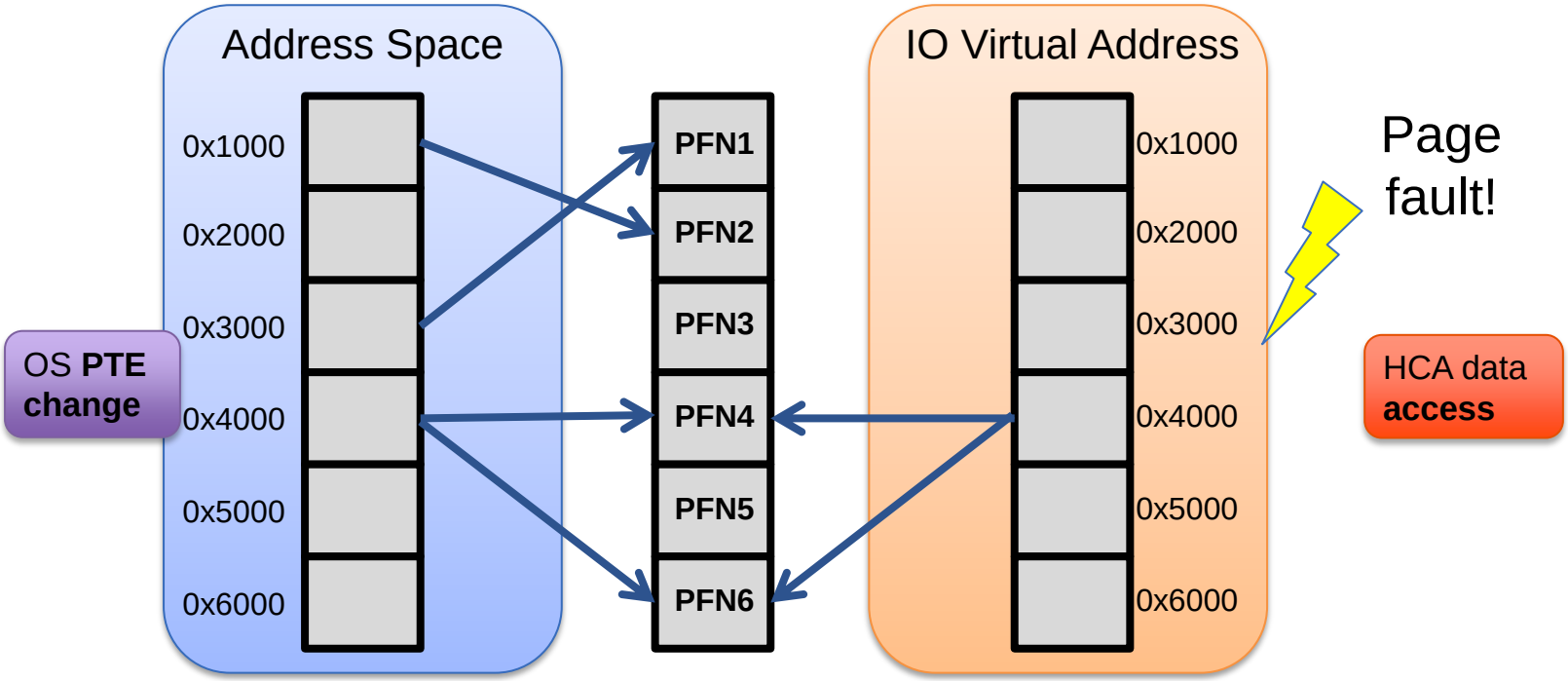
# Memory Registration Challenges

- Registered memory size limited to physical memory
- Requires special memory locking privileges
- Registration is a costly operation
- Requires careful application design for high performance
  - Bounce buffers
  - Pin-down caches
- Keeping address space and registered memory in synch is hard and error prone

# On Demand Paging

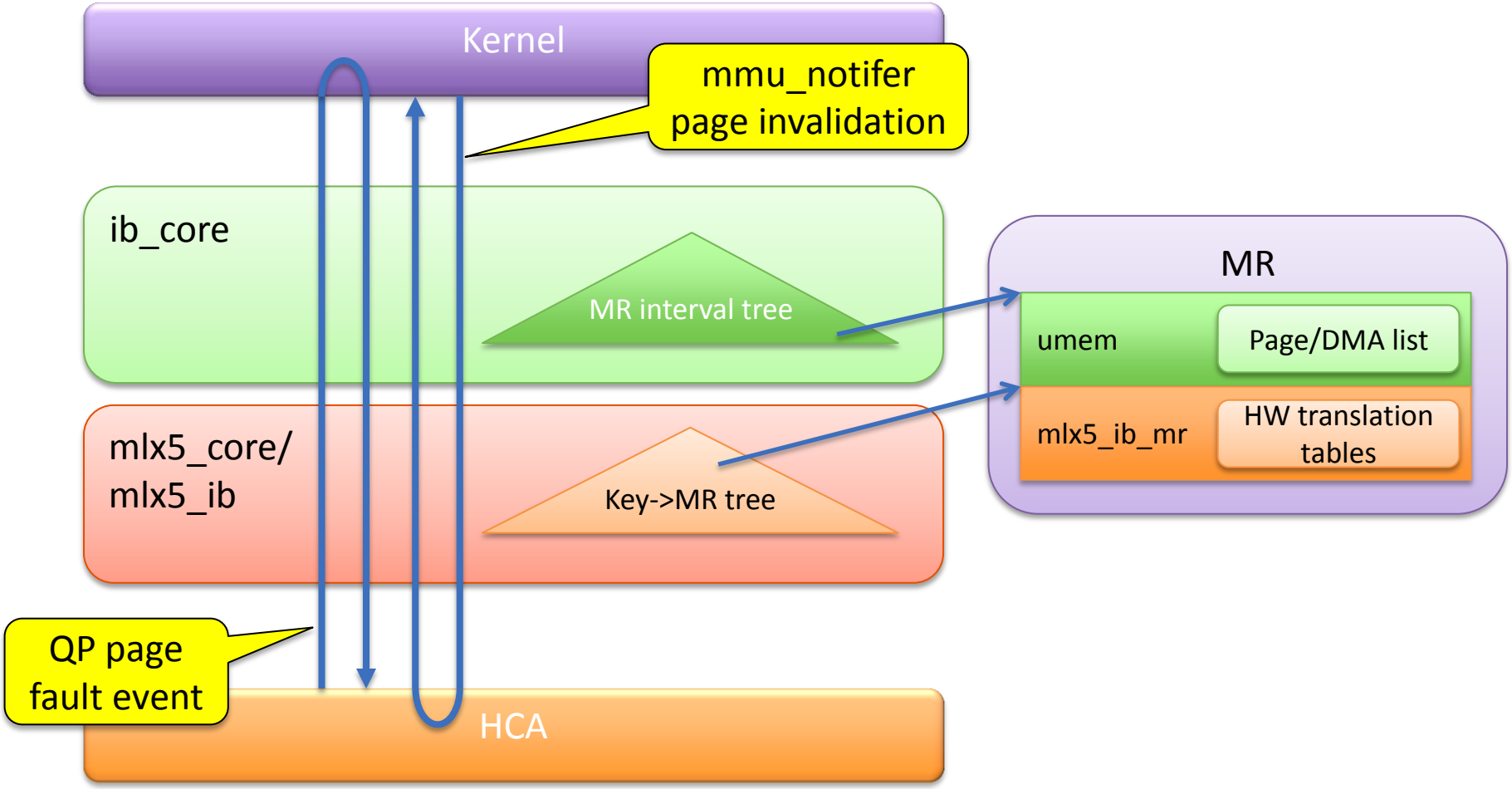
- MR pages are ***never*** pinned by the OS
  - Paged in when HCA needs them
  - Paged out when reclaimed by the OS
- HCA translation tables may contain non-present pages
  - Initially, a new MR is created with non-present pages
  - Virtual memory mappings don't necessarily exist
- Advantages
  - Greatly simplified programming
    - Reduce/eliminate registrations, no copying, no caches
  - Unlimited MR sizes
    - No need for special privileges
  - Physical memory optimized to hold current working set
    - For both CPU and IO access

# ODP Operation




ODP promise:
IO virtual address mapping == Process virtual address mapping

# Implementation



# ODP capabilities

```
enum odp_transport_cap_bits {
        ODP_SUPPORT_SEND     = 1 << 0,
        ODP_SUPPORT_RECV     = 1 << 1,
        ODP_SUPPORT_WRITE    = 1 << 2,
        ODP_SUPPORT_READ     = 1 << 3,
        ODP_SUPPORT_ATOMIC   = 1 << 4,
};

enum odp_general_caps {
        ODP_SUPPORT = 1 << 0,
};

struct ibv_odp_caps {
        uint32_t comp_mask;
        uint32_t general_caps;
        struct {
                uint32_t  rc_odp_caps;
                uint32_t  uc_odp_caps;
                uint32_t  ud_odp_caps;
                uint32_t  xrc_odp_caps;
        } per_transport_caps;
};

int ibv_query_odp_caps(struct ibv_context *context,
                       struct ibv_odp_caps *caps,
                       size_t caps_size);
```

# ODP Memory Regions

```
enum ibv_access_flags {
      IBV_ACCESS_LOCAL_WRITE          = 1,
      IBV_ACCESS_REMOTE_WRITE         = (1<<1),
      IBV_ACCESS_REMOTE_READ          = (1<<2),
      IBV_ACCESS_REMOTE_ATOMIC        = (1<<3),
      IBV_ACCESS_MW_BIND              = (1<<4),
      IBV_ACCESS_ON_DEMAND            = (1<<5)
};

struct ibv_mr *ibv_reg_mr(struct ibv_pd *pd, void *addr,
                          size_t length, int access);
```

- Registering the whole address space
  - ibv_reg_mr(pd, NULL, (u64) -1, flags)
  - Memory windows may be used to provide granular remote access rights

# Usage Example

```
int main()
{
    struct ibv_odp_caps caps;
    ibv_mr *mr;
    struct ibv_sge sge;
    struct ibv_send_wr wr;
    ...
    if (ibv_query_odp_caps(ctx, &caps, sizeof(caps)) ||
        !(caps.ud_odp_caps & ODP_SUPPORT_SEND))
            return -1;
    ...
    p = mmap(NULL, 10 * MB, PROT_READ | PROT_WRITE, MAP_SHARED, 0, 0);
    ...
    mr = ibv_reg_mr(ctx->pd, p, 10 * MB, IBV_ACCESS_LOCAL_WRITE | IBV_ACCESS_ON_DEMAND);
    ...
    sge.addr = p;
    sge.lkey = mr->lkey;
    ibv_post_send(ctx->qp, &wr, &bad_wr);
    ...
    munmap(p, 1 * MB);
    p = mmap(p, 1 * MB, PROT_READ | PROT_WRITE, MAP_SHARED, 0, 0);
    ...
    ibv_post_send(ctx->qp, &wr, &bad_wr);
    ...
    return 0;
}
```

# Memory Prefetching

- Best effort hint
  - Not necessarily all pages are pre-fetched
  - No guarantees that pages remain resident
  - Asynchronous
    - Can be invoked opportunistically in parallel to IO
- Use cases
  - Avoid multiple page faults by small transactions
  - Pre-fault a large region about to be accessed by IO
- EFAULT returned when
  - Range exceeds the MR
  - Requested pages not part of address space

```
struct ibv_prefetch_attr {
        uint32_t comp_mask;
        int flags; /* IBV_ACCESS_LOCAL_WRITE */
        void *addr;
        size_t length;
};

int ibv_prefetch_mr(struct ibv_mr *mr,
                    struct ibv_prefetch_attr *attr,
                    size_t attr_size);
```

# Statistics

- Core statistics
  - Maintained by the IB core layer
  - Tracked on a per device basis
  - Reported by sysfs

- Use cases
  - Page fault pattern
    - Warm-up
    - Steady state
  - Paging efficiency
  - Detect thrashing
  - Measure pre-fetch impact

```
/sys/class/infiniband_verbs/uverbs<dev-idx>/
        invalidations_faults_contentions
        num_invalidation_pages
        num_invalidations
        num_page_fault_pages
        num_page_faults
        num_prefetches_handled
```

| Counter name | Description |
|---|---|
| invalidations_faults_contentions | Number of times that page fault events were dropped or prefetch operations were restarted due to OS page invalidations |
| num_invalidation_pages | Total number of pages invalidated during all invalidation events |
| num_invalidations | Number of invalidation events |
| num_page_fault_pages | Total number of pages faulted in by page fault events |
| num_page_faults | Number of page fault events |
| num_prefetches_handled | Number of prefetch Verb calls that completed successfully |

# Statistics (continued)

- Driver debug statistics
  - Maintained by the mlx5 driver
  - Tracked on a per device basis
  - Reported by debugfs
- Use cases
  - Track accesses to non-mapped memory
  - ODP MR usage

```
/sys/kernel/debug/mlx5/<pci-dev-id>/odp_stats/
        num_failed_resolutions
        num_mrs_not_found
        num_odp_mr_pages
        num_odp_mrs
```

| Counter name | Description |
|---|---|
| num_failed_resolutions | Number of failed page faults that could not be resolved due to non-existing mappings in the OS |
| num_mrs_not_found | Number of faults that specified a non-existing ODP MR |
| num_odp_mr_pages | Total size in pages of current ODP MRs |
| num_odp_mrs | Number of current ODP MRs |

# News

- Connect-IB Support
  - Initially UD and RC
    - DC will follow
  - Address space key for local access
- Initial testing with OpenMPI
  - No more memory hooks!
- Release planned for MLNX_OFED-2.3
- ODP patches submitted to kernel

# Thank You

OPENFABRICS ALLIANCE

10TH ANNUAL INTERNATIONAL DEVELOPER WORKSHOP