eXtended Reliable Connection (XRC)

Dror Goldenberg
Mellanox Technologies

Agenda

- Introduction
- XRC API
- Status

What Has Changed?

Was:

Scalable **R**eliable **C**onnection – **SRC**

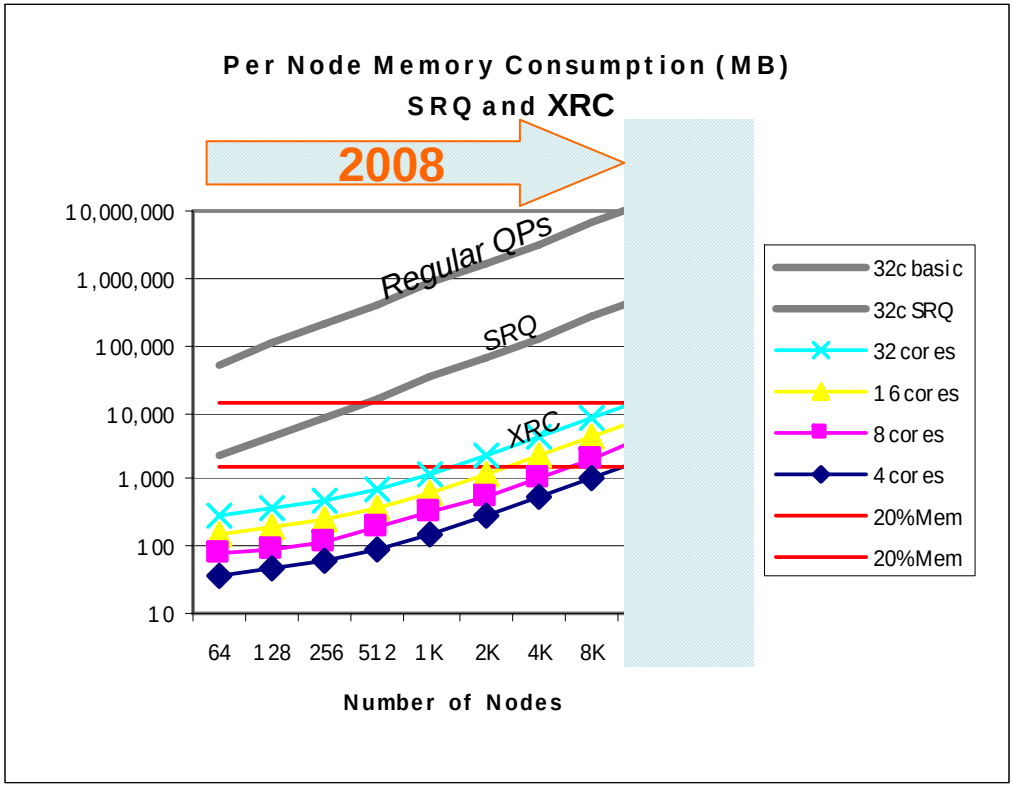
Is:

e**X**tended **R**eliable **C**onnection - **XRC**

Motivation - Why XRC ?

- For MPI-like applications
- QPs per node*
  - QPs = Nn x Nc

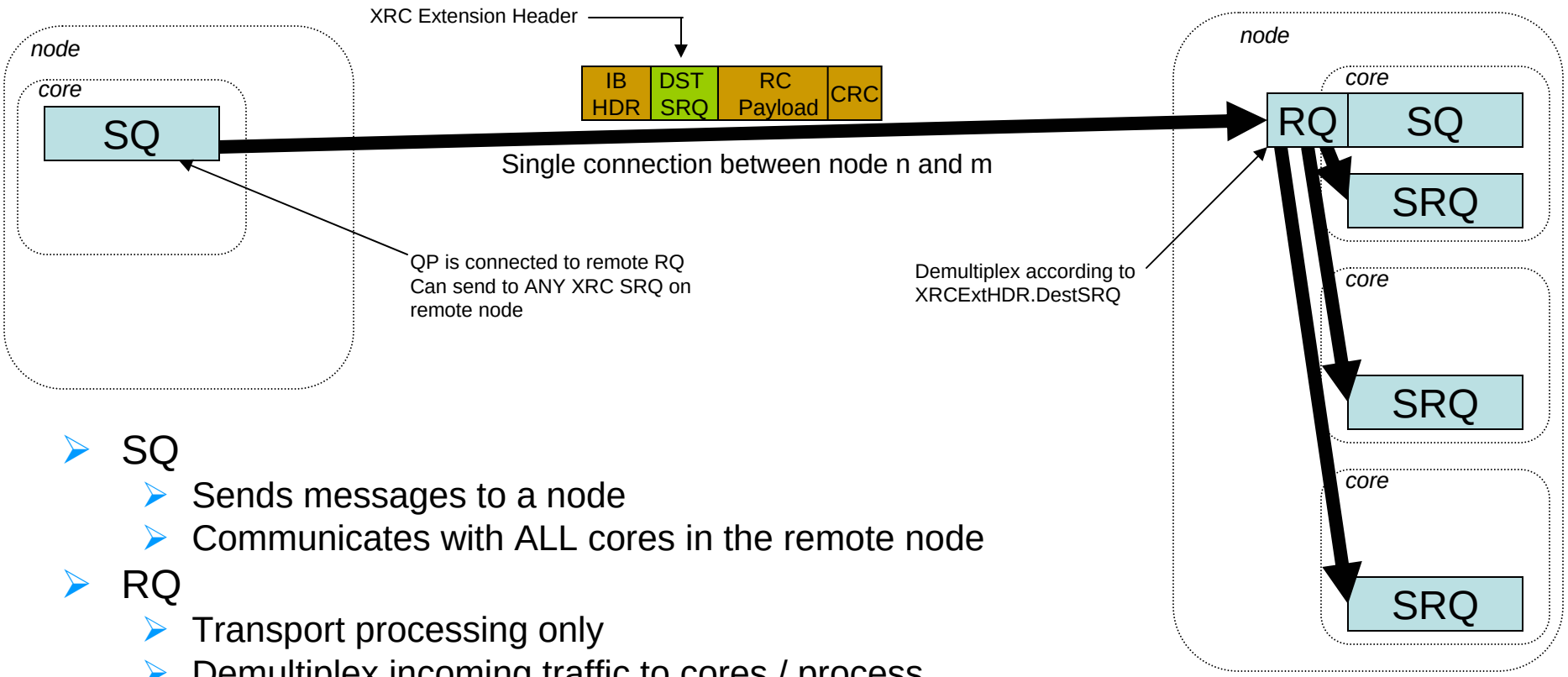
    Down from Nn x Nc^2
- Memory per QP
  - Send WQ buffer ~32KB
- Memory per core
  - SRQ data buffers ~7680KB
- Comfort zone
  - ~20% out of ~2GB/core



eXtended Reliable Connection reduces memory footprint
and is essential for scaling beyond ~4K nodes / 4 cores

* Nc = CPU Cores per Node
Nn = Nodes

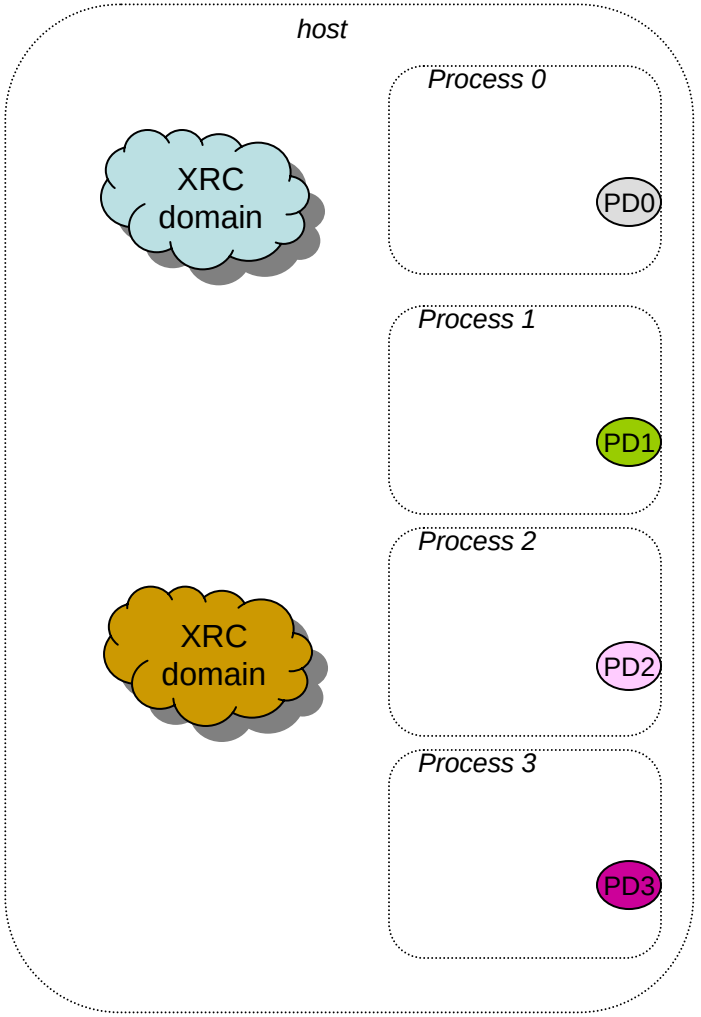
eXtended Reliable Connection



- SQ
  - Sends messages to a node
  - Communicates with ALL cores in the remote node
- RQ
  - Transport processing only
  - Demultiplex incoming traffic to cores / process
    - Incoming Sends – take WQEs, CQ and PD from SRQ
    - Incoming RDMA Read/Write – take PD from SRQ
  - XRC domain validation
- Similar concept as IBTA RD

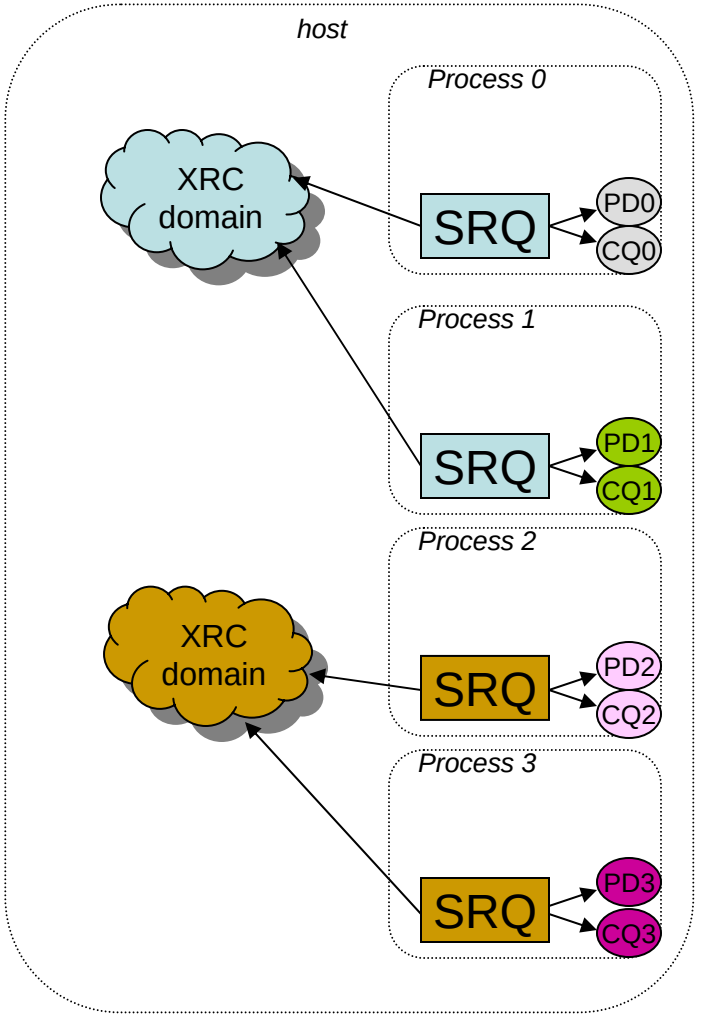
XRC Domain Creation

- Ibv_open_xrc_domain()
  - Creates/attaches to XRC domain
  - XRC domains
    - Associated with inodes
    - XRC domain sharing is like file sharing
- Ibv_close_xrc_domain()
  - Close/detach from XRC domain
- Requires device capability
  - IB_DEVICE_XRC
  - Support: ConnectX



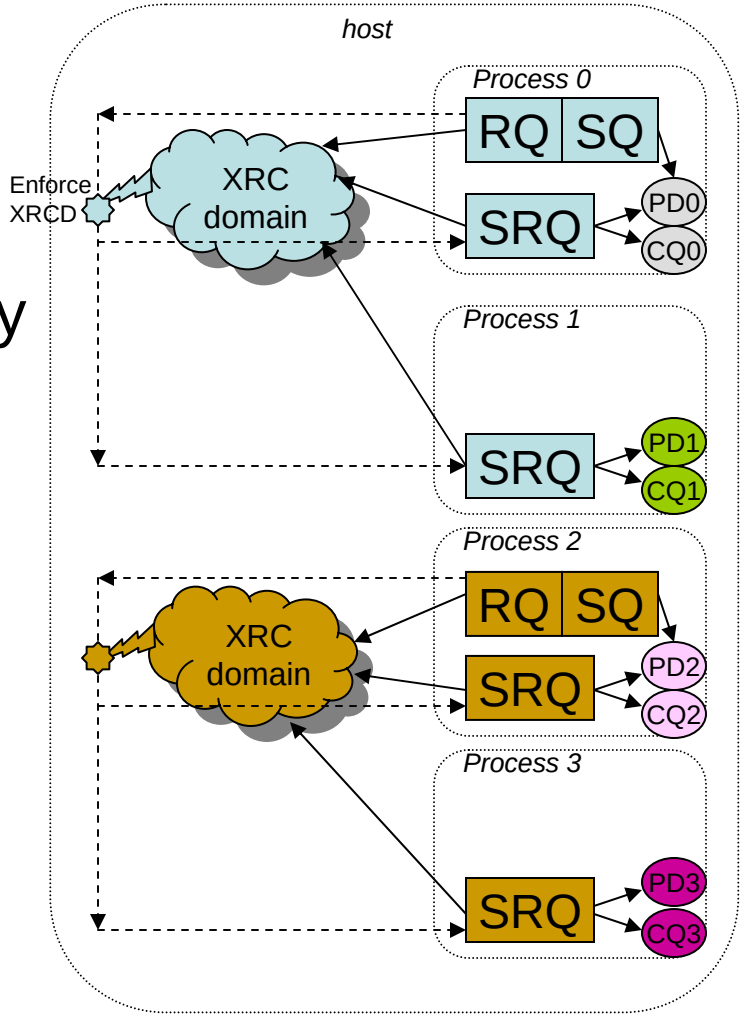
Attach SRQs to XRC Domain

- Ibv_create_xrc_srq()
  - XRC Domain
    - SRQ can receive messages from QPs that are on the same XRC domain (not necessarily in the same process)
  - PD
    - Used for memory accesses through the SRQ (RDMA/Send)
  - CQ
    - Used to complete incoming Sends



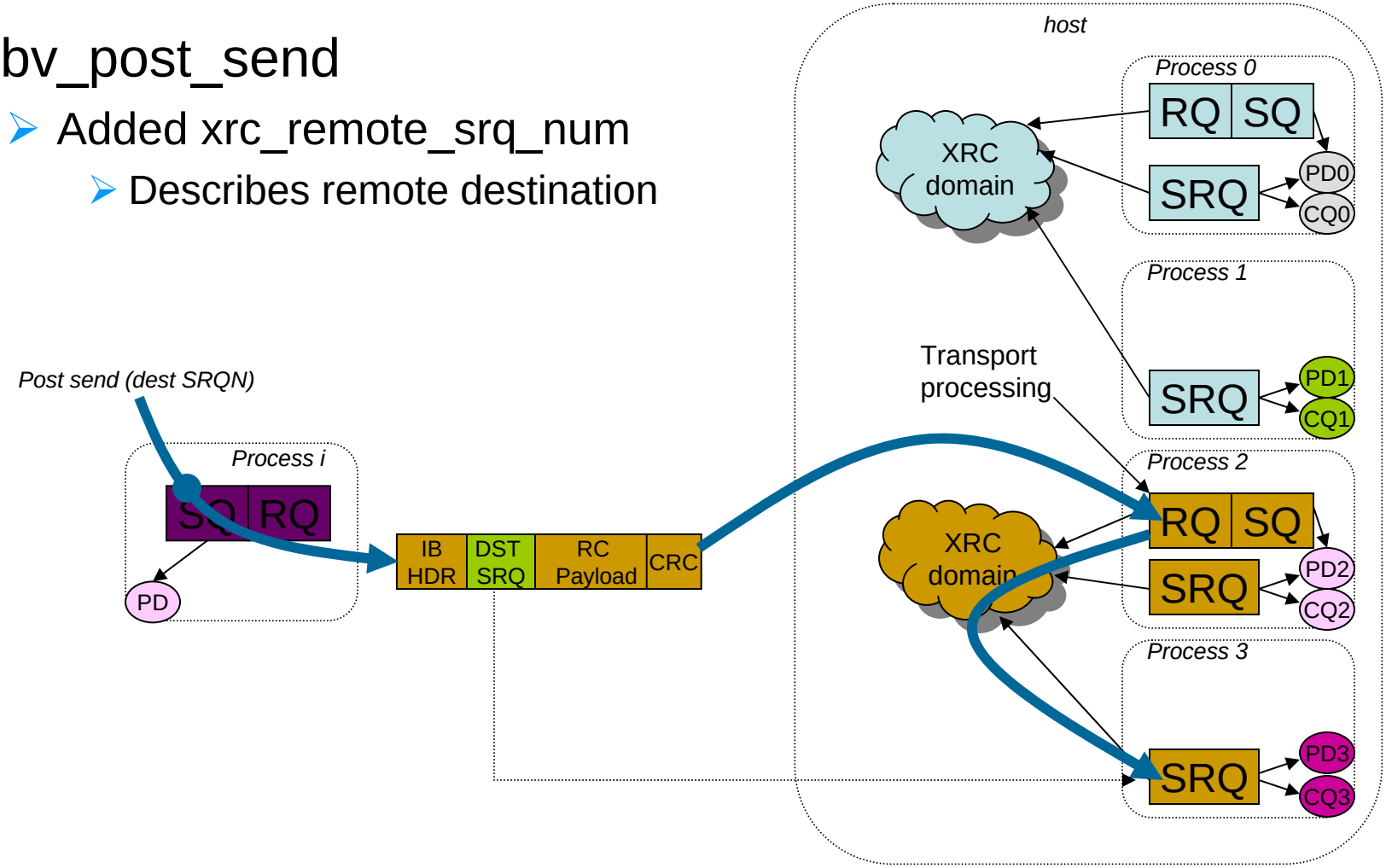
Attach QP to XRC Domain

- Ibv_qp_init
  - transport service: IBV_QPT_XRC
  - added ibv_xrc_domain
- RQ can deliver messages to any SRQ on the same XRC domain
  - RQ runs the transport and demultiplexes incoming messages
  - RQ.PD is meaningless



Post Send

- Ibv_post_send
  - Added xrc_remote_srq_num
    - Describes remote destination



XRC Status

- IB core and mlx4 support
  - User and kernel mode
- Integrated into OFED 1.3
- Mainline kernel integration - under review