2022 OFA Virtual Workshop

BRIDGING RDMACM TRAFFIC BETWEEN INFINIBAND AND ROCE

Christoph Lameter, Ph.D., Senior IT Experte

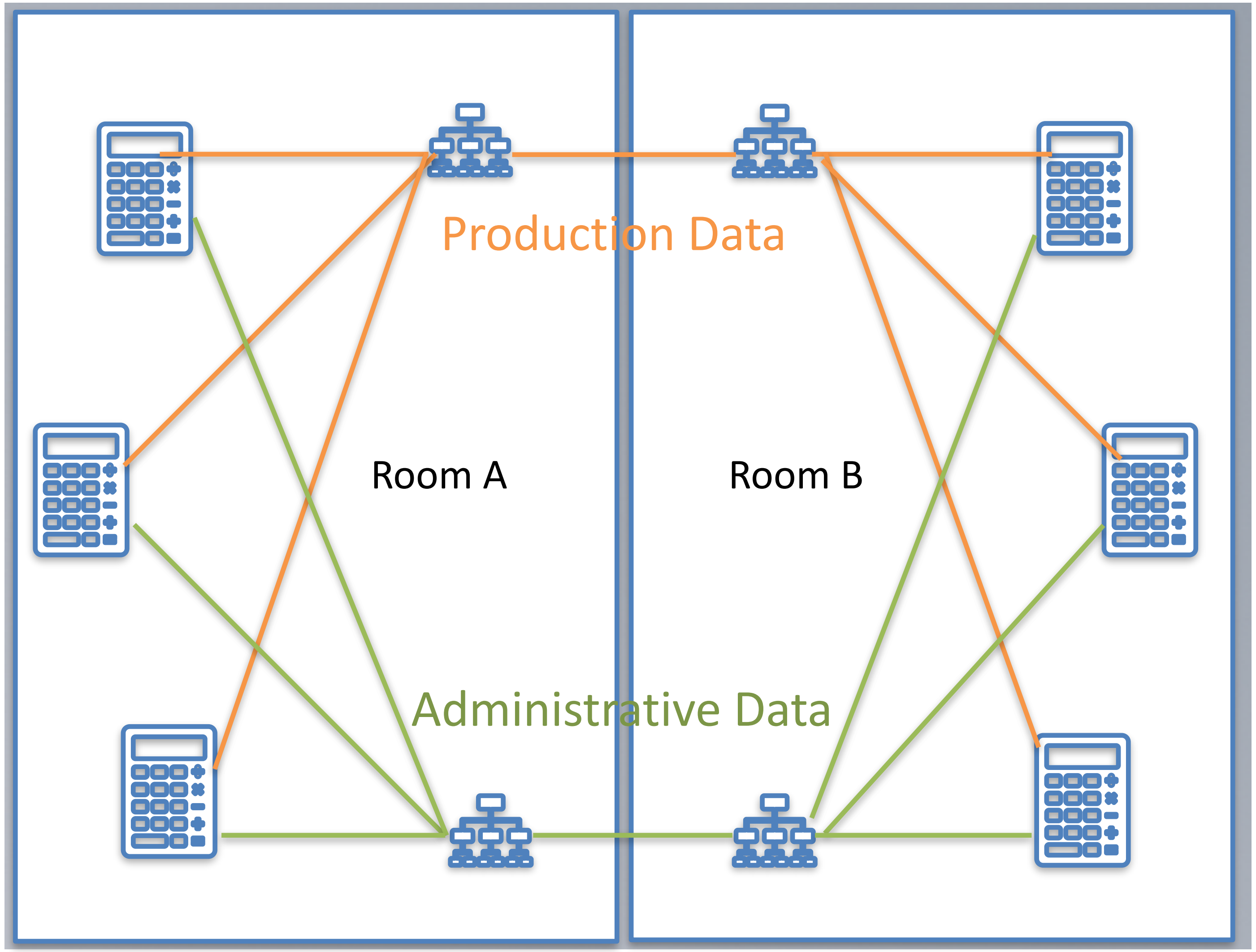
cl@linux.com 26.April 2022

MIGRATION FROM INFINIBAND TO ETHERNET

FROM INFINIBAND TO ETHERNET RDMA

Migrating a large Infiniband installation without impacting operations

- Infiniband fabric with ~300 nodes with redundant links
- We intend to perform a gradual migration via a special gateway software
  - Unicast (low volume)
  - Multicast (high volume)
- Confinity LLM Middleware
- Method of migration
  - Establish redundant gateways
  - Move a single host or multiple to the “other” side.
  - Similar basic Architecture after migration is complete



SINGLE SERVER CONFIGURATION

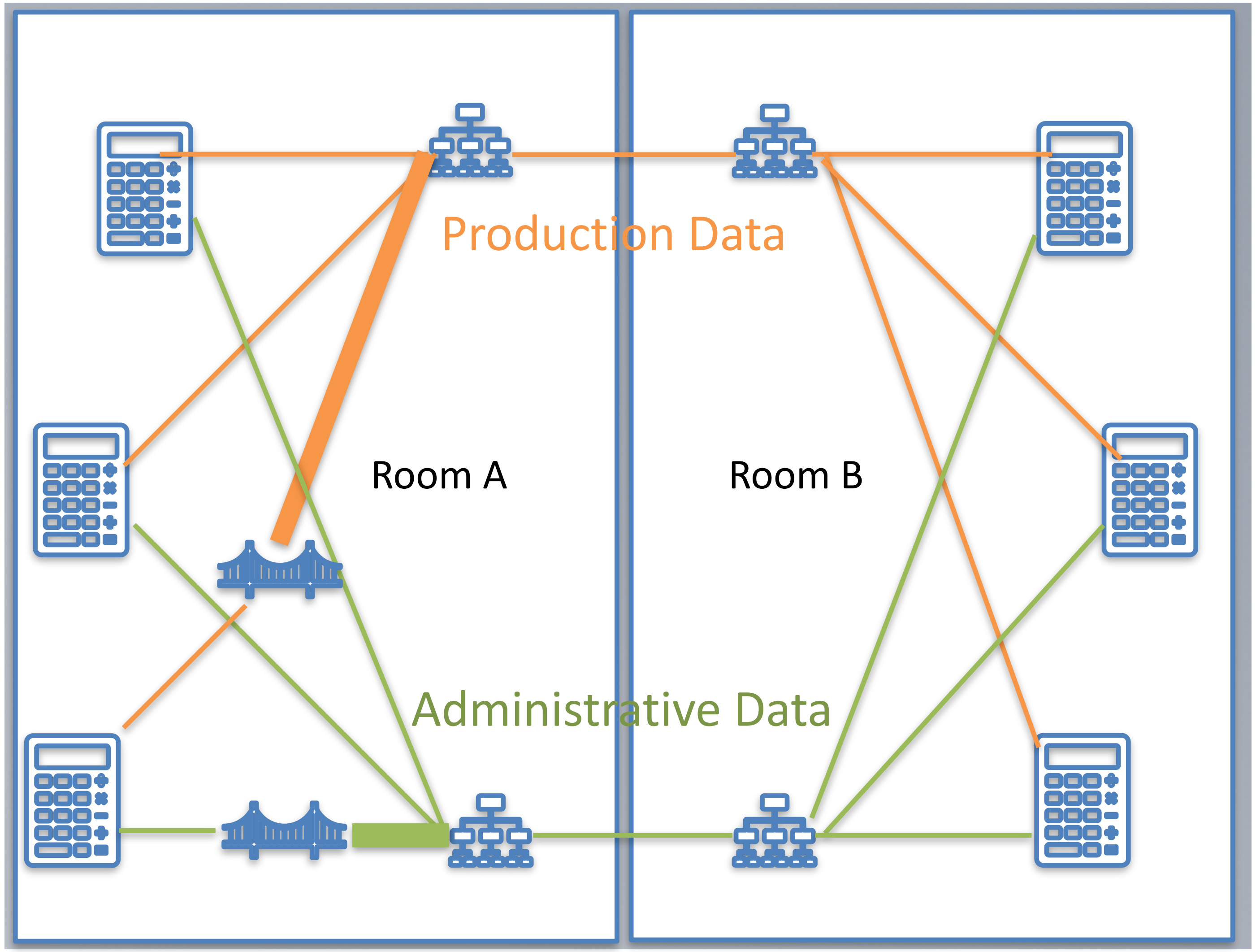
- Typical Network Links
  - RDMA Production Data (QDR INFINIBAND -> 25G ROCE)
  - RDMA Administrative Data (QDR INFINIBAND -> 25G ROCE)
  - Access LAN (2x10G Ethernet)
  - Market Data (10G Ethernet)
  - Order Data (10G Ethernet)

- Migration is needed primarily because of the age of the QDR fabric.
- “Aging” issues are mostly due to the limits of vendor support. The fabric speed is sufficient and always has been.
- Choice is 25G ROCE due to similarity to 10G. Both are using SFPs and allow easier handling.
- SFP Cabling is widely used.
- QSFP has thick cables and is more specialized.
- Volume for traffic is limited. Conversion is realistic.
- Links can be a multiple of 25G if necessary. If volumes are high a server can utilise 50G or 100G links.

INTERMEDIATE BRIDGED CONFIGURATION

Temporary concurrent operation of ROCE and Infiniband

- Temporary Bridges are added with 100G upstream links to each of the two fabrics
- Bridges have a 100G Ethernet switch on the ROCE side (not shown for brevity) that on the ROCE side link to nodes with 25G links.
- Initially 2 Ethernet switches and 2 Bridges are needed to provide the ability to connect systems to either Fabric in one room.
- Ultimately all 4 Infiniband switches will get paired with 100G Switches to allow alternate fabric use in both rooms.
- Traffic can then flow on either ROCE or INFINIBAND between arbitrary nodes.
- Infiniband Switches can later be decommissed if a room has no need for Infiniband links anymore.

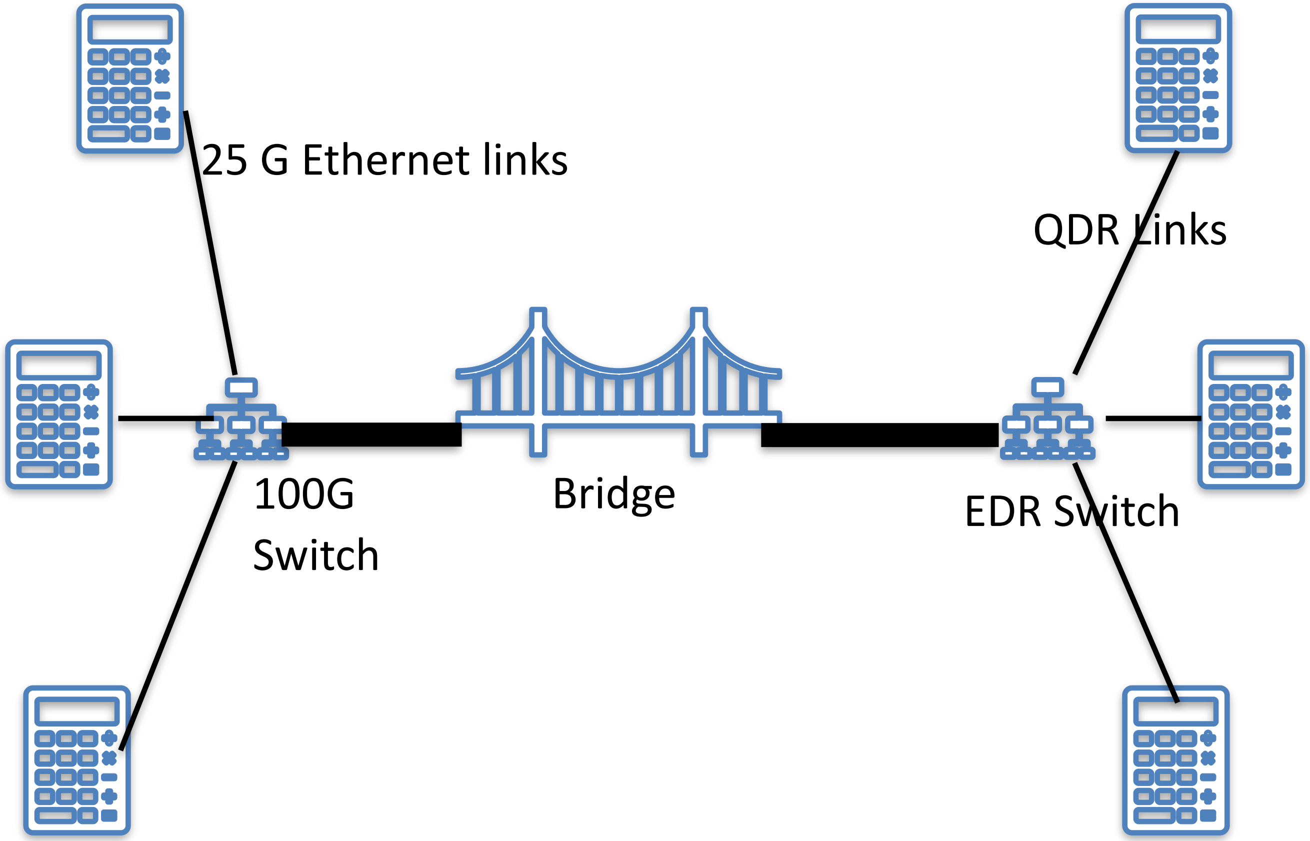


BRIDGE

BRIDGE HARDWARE

Connectivity and Switches

- Bridge
  - A regular 1U server
  - ConnectX6 Dual NIC
    - 1x 100G/200G link to Ethernet Switch
    - 1x EDR/HDR link to Infiniband Switch
- Ethernet Switch
  - 25G Links for typical servers
  - Higher speeds for special needs
- Infiniband Switch
  - QDR links to typical servers
  - EDR / QDR to legacy switches



BRIDGE SOFTWARE

Sample subtitle here

- **Unmodified Redhat 8 Enterprise Linux**
  - **Inbox drivers**
- **Uses ib2roce bridging software developed as a open source project on Github and intended to be part of the rdma-core subsystem.**
- **Ib2roce supports RDMACM based bridging**
  - **Multicast**
  - **Unicast**
  - **Multiple ib2roce instances can coordinate and provide redundant connectivity [incomplete].**
  - **Diagnostics for unicast and multicast connections**
  - **Control over failover [not implemented yet]**
  - **Control over multicast message rates**

IB2ROCE operates through 3 Queue Pairs open on both Fabrics:

- Multicast QP: Used to receive and send multicast Traffic

- Unicast QP: Used to multiplex unicast traffic of RDMACM connections

- Control QP: Used to process unicast connection requests

MULTICAST TRAFFIC

MULTICAST VIA THE GATEWAY

Almost all application traffic

Multicast on ROCE is different than on Infiniband. Infiniband can slow down senders through back pressure. Therefore multicast packets will not be dropped on Infiniband.

However, flow control on ROCE was only implemented for unicast connections. It is therefore more likely that multicast packets will get dropped and it is not possible to saturate the full bandwidth of the links with multicast traffic like under Infiniband.

Therefore, the rates of traffic to the endpoints have to be controlled in such a manner that the application can safely process the data.

Another issue is that the links to the hosts have a lower bandwidth than the links from the bridge to the switches. The bridge can only send a fraction of its own bandwidth to a particular endpoint or multicast group otherwise the switch will have to drop multicast packets. Therefore ib2roce must include control mechanisms to limit the traffic rate to the nodes in the fabric and needs to use its memory to buffer messages if those traffic rates are exceeded.

- Trivial approach: Subscribe to multicast groups on both sides and copy message contents unchanged.
- RDMA stack does the header conversion between Infiniband and ROCE.
- A single thread can bridge 100G Multicast traffic.
  - Typically packet size is around 4k
  - IB2ROCE can bridge 1.4 Mio pps on a single thread with NOHZ_FULL from user space
- One problem is that hosts can be overrun because the bridge can send too fast for the receivers
  - Middleware can only handle 100k .. ~400K pps depending on application load.
  - Options to enable throttling via “rate” specification in the RDMA global routes.
  - Software implementation of a rate limiter. Allows the burst of N packets and then slows down.

UNICAST TRAFFIC

UNICAST BRIDGING

Forwarding RDMACM unicast streams between Infiniband and ROCE

In order to use the native RDMA unicast forwarding mechanism the RDMA subsystem in the kernel needs to be patched. This involved 3 different changes:

a) Implement the ability to redirect MAD requests to gateways to a different QP

b) Implement that the response to SIDR REQ (the SIDR REP) goes back to the originating QP# that issued the SIDR REQ)

c) The private data for SIDR REQ/ REP must contain the originating Source QP# in addition to the existing information about the target IP addresses for the RDMA CM endpoint

- Trivial approach: CLLM was modified to use the socket API for unicast. In low volume situations the kernel can route the traffic. However, the traffic rate supported by the kernel is limited to at around 100kpps and the latency is increased more than ten times.
- A more sophisticated approach: The gateway performs MAD processing of SIDR REQuests and SIDR REPlys that are used to resolve the port numbers to QP numbers and then establishes a forwarding between RDMACM QPs on both fabrics.
- RDMA allow the redirection of SIDR requests via the kernel routing tables to a gateway (undocumented but it strangely works because ARP is used for IPv4 resolution). The MAD packets will flow through the gateway and can be modified on the fly.
- The RDMA subsystem receives MAD traffic on QP1. So a kernel patch allows the redirection of SIDR requests to an alternate QP# where the gateway can receive SIDR requests, modify them and forward them onto the other fabric.
- The gateway opens a unicast QP on both Ethernet and IB and uses tables to forward packets between both fabrics. Port resolution was done by the SIDR requests. Unicast forwarding is occurring between QP# on either side of the gateway. The RDMA stack is used for the header conversion by receiving and resending the payload.
- The unicast performance is similar to the multicast case and allows more than 1 mio pps. However, the payload is usually smaller so it is not possible to saturate the links with unicast traffic.

GITHUB IB2ROCE PROJECT

IB2ROCE OPENSOURCE PROJECT

The gateway between ROCE and Infiniband

- **Hosted on github**
  - Rdma-core extension https://github.com/clameter/rdma-core
  - Kernel patches https://github.com/clameter/linux
- Implemented in a new directory in the rdma-core package
- The source code produces working binaries but is in need of additional work
  - Proper Documentation
  - The handling of the QP redirection for unicast forwarding is a bit awkward since the QP# that has to be set is only known after ib2roce has started.
  - The kernel patches may need some better ideas to properly integrate with the RDMA subsystem.
  - There is additional code in the source that may not be needed anymore
    - RAW packet sockets to avoid the kernel patches
    - Some RDMACM handling from earlier revs.

OPENFABRICS ALLIANCE

2022 OFA Virtual Workshop

THANK YOU

Christoph Lameter, Senior IT Experte

cl@linux.com

DEUTSCHE BÖRSE GROUP