2021 OFA Virtual Workshop

Infiniband Developments and the T7 Trading Engine

Christoph Lameter, Ph.D., Senior IT Experte

Deutsche Boerse AG

Legalese

- All of the information provided here is about experiments and analysis done in the Lab at the Deutsche Boerse. This does not mean that any of these situations occur in real production system or that any of features discussed here are in actual use or will be or ever have been.
- The presentation here only reports on research done in a lab.
- The actual details of the T7 production system are out of scope of this talk and confidential.
- T7 system characteristics mentioned here where discussed in the official publications of the Deutsche Boerse in documentation related to the use of the T7 system by third parties.
- Please refer to the published official documentation on the Website of the Deutsche Boerse for technical information about the T7 Trading system.

The T7 Trading Engine and Infiniband Technology

T7 Characteristics

- The Deutsche Boerse has been running Infiniband reliably in a production environment for almost a decade now. Deutsche Boerse has become one of the largest Stock Exchanges and runs a multiplicity of financial market places among them EUREX and Xetra. These are the largest trading venues in Europe.

- T7 core systems operate using Infiniband and the deployment of a high speed fabric as used in High Performance Computing is unique. T7 was designed for stability, reliability and failover first and then for performance instead of making optimal performance the primary goal.

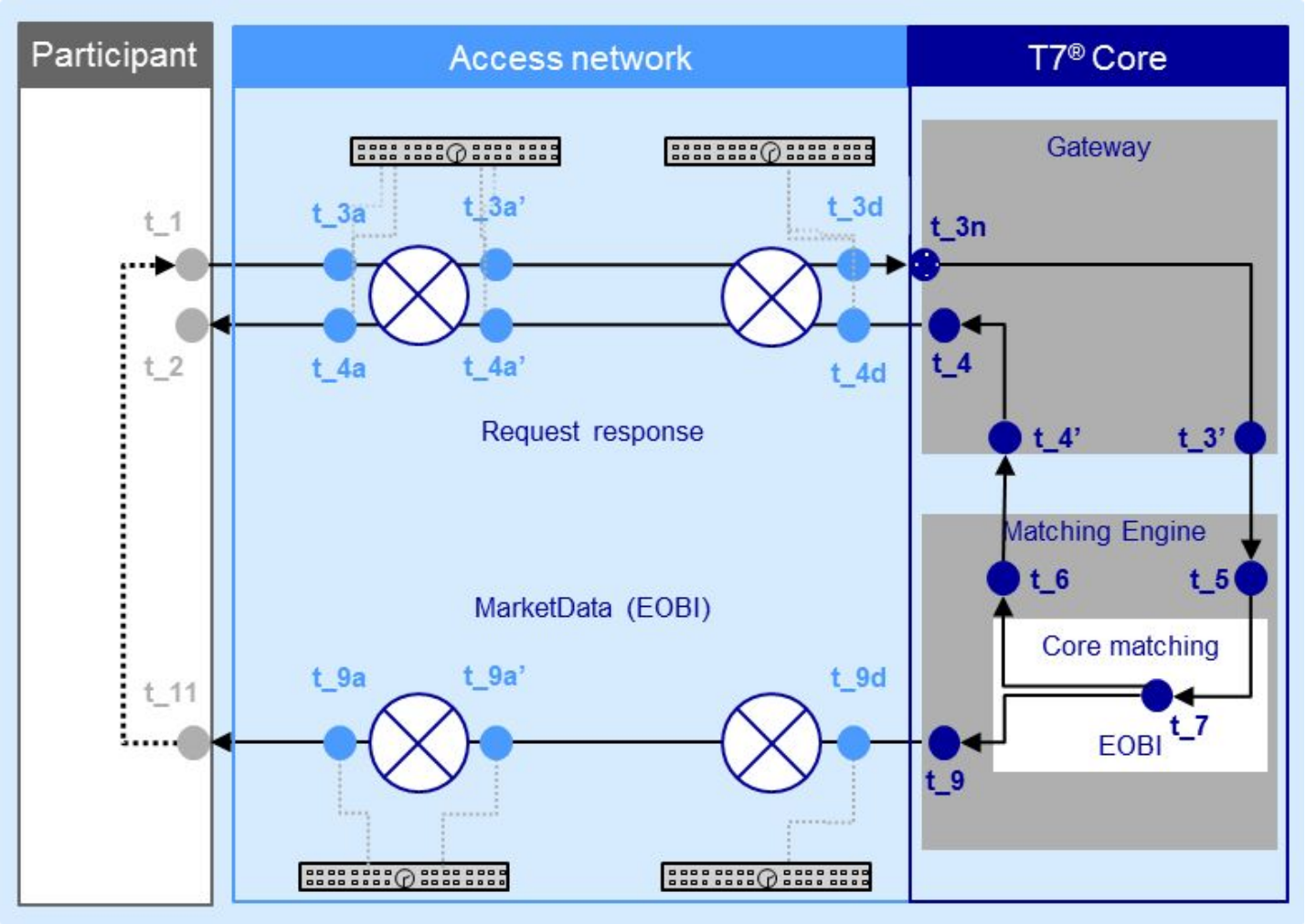
- Customers connect to T7 mostly through 10G Ethernet and for that purpose T7 has a Ethernet based fan-in and fan-out design.
- Competition for low latency trading on the 10G Ethernet Infrastructure is intense and has reached a situation where nanoseconds make a difference if a company wants to engage in the most lucrative trading opportunities.
- The T7 trading systems uses middleware to isolate the application from the details of the RDMA stack. The middleware provided by Confinity Solutions is called "Low Latency Messaging". CLLM was initially developed by IBM and implements reliability and failover of services on top of RDMA.

CONFINITY SOLUTIONS

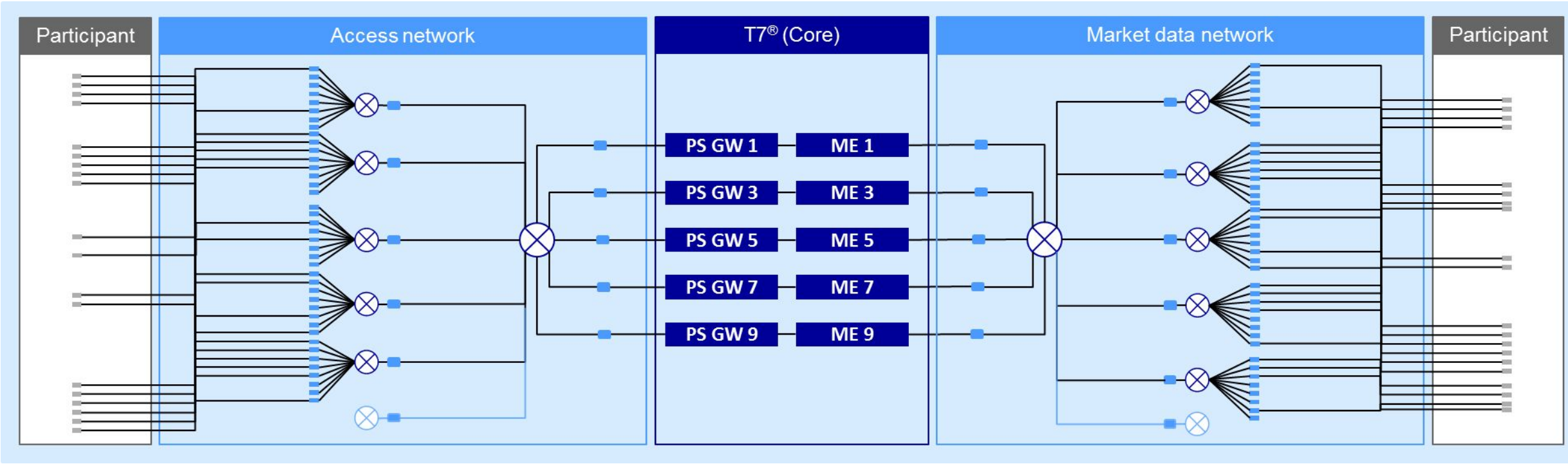
Ethernet Interface for Customers

A typical communication flow

- **Fan-in: Orders (Buy/Sell) flow into the T7 core via a set of switches.**
- **Fan-out: Marketdata (Trading events f.e.) flow out of T7 back to the Customer.**
- **Customer decides based on Events (including Marketdata) if to issue orders and to purchase or sell instruments on the markets.**
- **Orders get matched and new marketdata is produced.**



Ethernet Fan in and Fan out

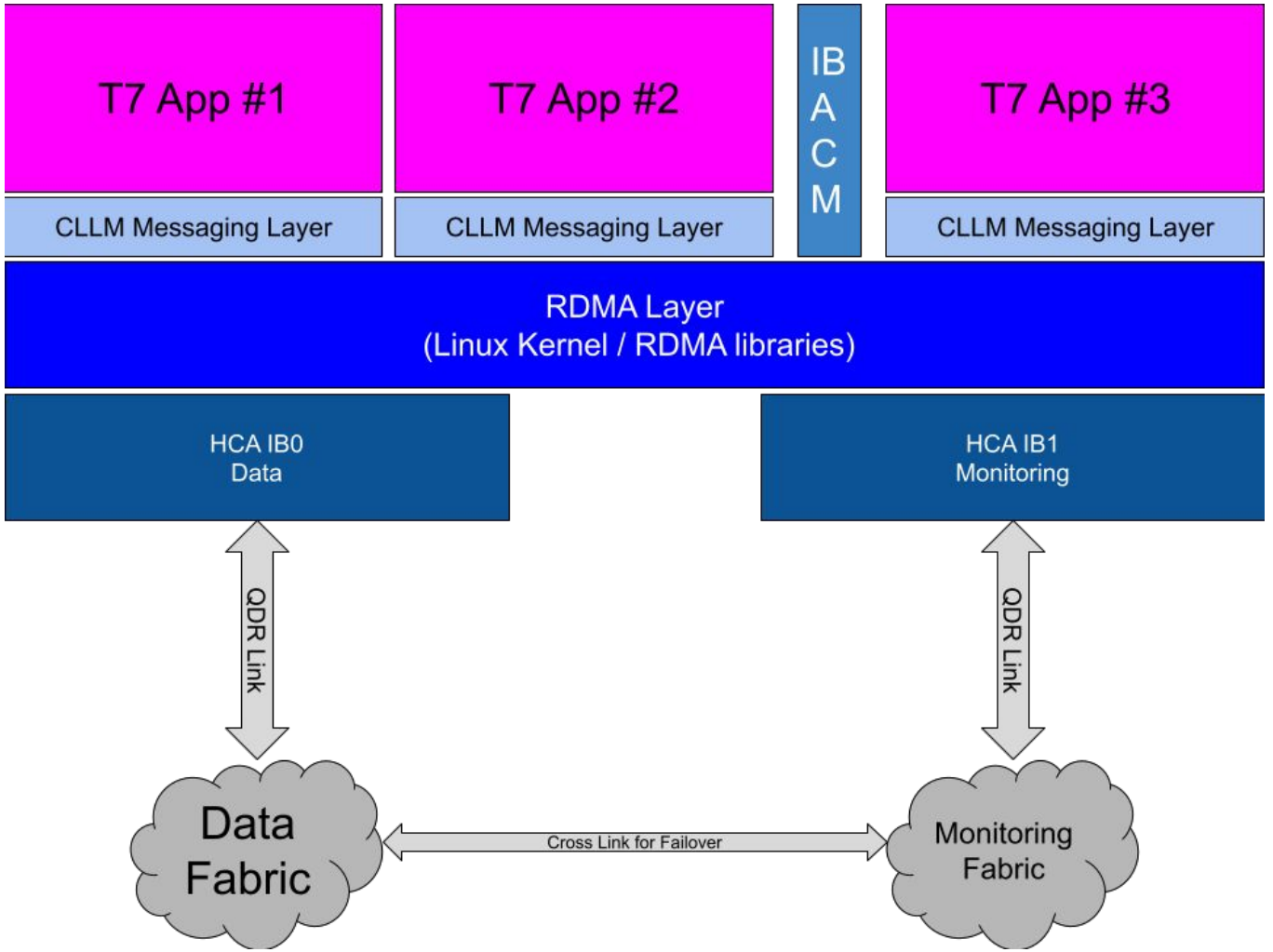


Fan in concentrates traffic that is then delivered to the Gateways

Fan out replicates traffic via multicast to customers.

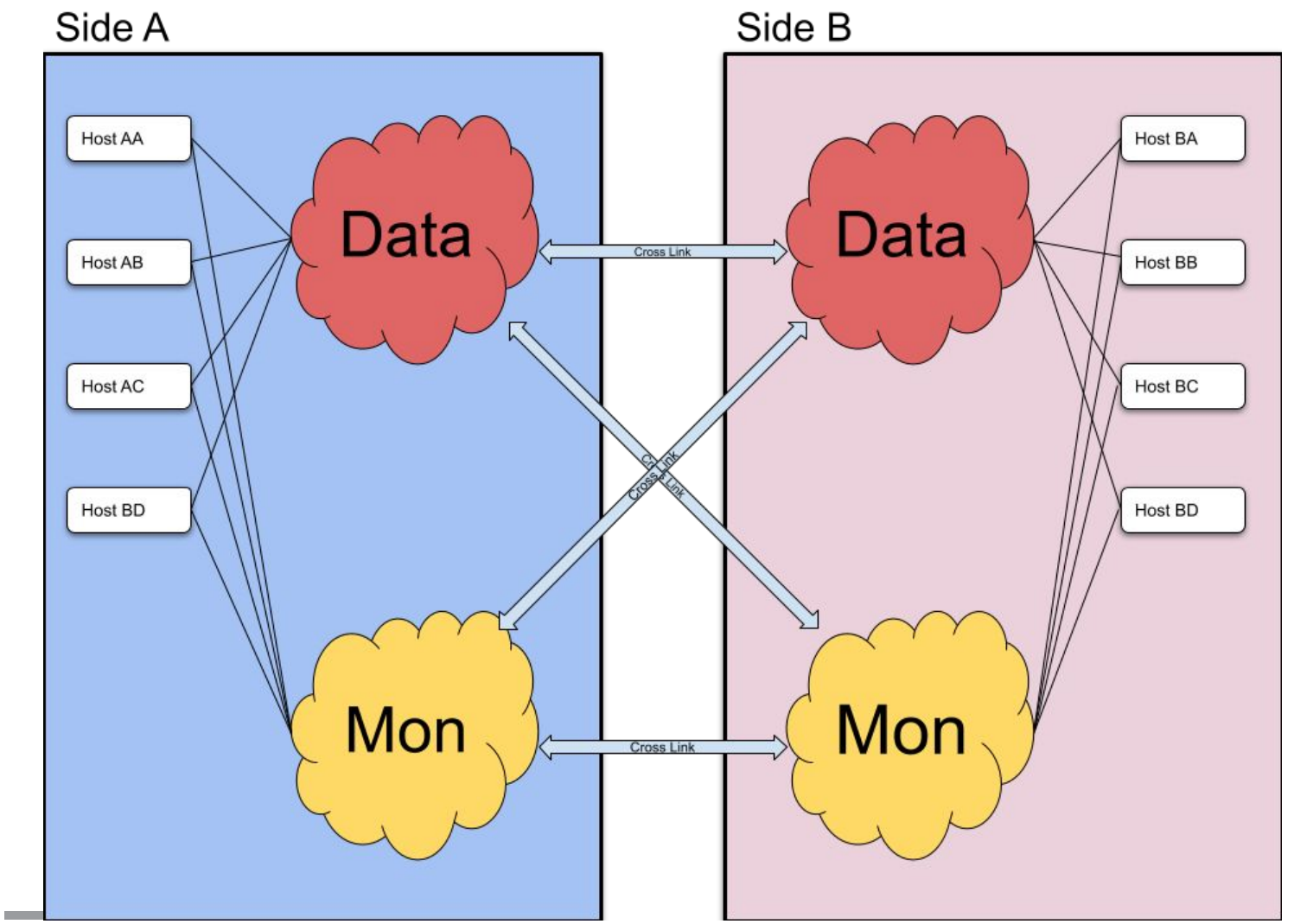
Hardware and Software Stack of a single Node

- A pretty standard Intel two node Enterprise class server
- Two Fabric connections allowing failover
- Hardware (Intel CPU/PCIe3) has been stable for a decade
- Multiple instances of CLLM operate on a machine. One per App.
- Multithreaded environment that is not designed according to a low latency design
- Holdoffs on QP processing in the range of milliseconds possible.
- An optional IBACM can avoid SM load.



Redundant Design of the Infiniband Fabric

- **4 Sections of Infiniband**
  - 2 Sides
  - 2 Fabrics with a different purpose
- **Two different physical locations**
- **No single point of failure**
  - A location may fail
  - Data or Mon Fabric may fail
  - A host may fail
  - A service on a host may fail
  - An arbitrary connection / link may fail



Multicast Use and Improvements

Multicast

- Multicast is used to spread information rapidly to a number of subscribers to that information. Replication of information is done by the switches in IB fabric. The sender can send one message to reach multiple subscribers.
- Multicast is used to implement redundancy as well. Multiple consumers can process the same information since a multicast group is joined as an address that does not map to a single system.
- The overwhelming majority of traffic on the fabric on the T7 system is therefore multicast and the data volume that can be processed by the T7 system is dependent on the effectiveness of processing multicast data and on the volume of data on the fabric.

- CLLM implements separate consumer and publisher instances. The problem is that the joins to the multicast groups were not setting the join attributes correctly. All instances were simply joining the multicast group and therefore publishers were receiving all data from the multicast group.
- Working with CLLM, we got a new version that implemented the sendonly join option so that datagrams were no longer unnecessarily send to the publishers. That measure reduced the volume of traffic in our lab to about a fourth.
- We then found that a bug existed during failover (ReRegistration processing in the kernel) and I submitted a patch to fix it. This is upstream in Linux 5.12 and is being integrated into RHEL 7 and 8.
- Given the high rates of multicast traffic the T7 application had workarounds to use unicast instead of multicast to avoid creating high volume of traffic. With the reductions it was possible to revert to the earlier use of multicast which caused further reduction in the volume of traffic and reduced the number of resolution requests to the subnet manager in our lab.

Subnet Manager

Subnet Manager

- The DB has a history of running the simple *OpenSM* in Redhat and running straight drivers from Redhat RHEL for an extensive time period (about a decade) without problems. There is a high level of trust in the Open Source drivers and SM. So that is where we started from.
- We saw in the lab an unusual high number of SM requests during startup of the T7 app and also during experiments with failover.
- In particular one issue was that we saw heartbeat retries during high volumes of traffic from the backup SM. So the volume of SM requests was reaching a critical level. We could not find a way in the *Redhat SM* and *Mellanox SM* to allow the placement of the heartbeat outside of the Infiniband Fabric. This was disconcerting issue because at a high level of SM traffic this could lead to a split brain issue where both primary and secondary SMs feel that they are in charge.
- We therefore did some experiments with osmtest (part of the RDMA diagnostics) where we attempted to intentionally saturate the fabric with SM requests to see what happens to a fabric in the worst case. To our surprise the whole system began to fail. Individual hosts started to reboot without discernible cause. We deduced from that that we certainly have a need here to keep the level of SM traffic under control and monitor the load on the SM carefully. VL15 traffic has no backpressure and overloading the fabric causes a high level of drops of Infiniband management packets which can be a serious problem.
- We contacted Mellanox to talk about the SM but were told that the openSM in Redhat is no longer maintained and it is advised not to use the SM from the github project either. The *Mellanox SM* is recommended as delivered with MOFED. However, the Mellanox SM still has not the full functionality. Mellanox suggested buying their *UFM Fabric Manager* that has the capability to configure the heartbeat connection to go through another medium for ultimate reliability and it has numerous improvements that are not available in the other SMs.

CLLM Middleware and the Subnet Manager

- Analysis of CLLM revealed that the *establishment of a unicast connection* requires multiple SM request. There is no caching of SM information in the RDMA subsystem for unicast connections. This only exists for Multicast. CLLM retries these requests at regular intervals. If a service on the IB fabric is not up then a continuous stream of traffic to the SM is created for each connection attempt. If a popular service that has multiple connections is terminated then a high sudden load of SM traffic can be created and that will persist until the service is started again.
- Failover. During failover to the secondary SM the SM will send out *REREG requests* so that all nodes re-register their endpoints. We saw maximum SM load at this point in particular since the T7 app had a huge number of unicast connections and all the connections had to be reestablished. The level of SM requests looked barely manageable by the SM.
- Requests from CLLM are mostly for information that was retrieved before so a way to cache the SM information would solve the problem. There are two solutions:
  - Use of the IBACM cache. IBACM keeps the SM information for a day by default. If we ensure that the mapping of GUID to LID does not change then this seemed safe to us.
  - Complete the implementation of rdma_getaddrinfo() to use the IPoIB cache for SM information.
- The severity of effect of the REREG requests to CLLM led to more talks with Mellanox. As a result a new feature was added to UFM which is called the STATIC SM LID feature which avoids having any reregister events at all as long as the GUID to LID mapping stays stable. The SM lid will not even change during a failover. This remains to be tested.
- It is advisable for CLLM to change its implementation of unicast handling to avoid repeated requests to the SM. We worked with CLLM engineers to deal with some other issues in connection handling but this is still outstanding.

Resilience and Risks of T7

Aspects of Resilience

- The basic T7 philosophy is that *any component can fail*
- CLLM implements *service level failover* as well as *failover between multiple HCA*
- Infiniband implements *failover for critical components* like the SM.
- Risks in this scenario because the SM was not evolving with the other components since the vendor converted it to a proprietary product.
- Risks because the Infiniband Interface of CLLM was not evolving. The company was focused on software features.
- Risk arise as because of the mitigation of one issue (multicast data volume) which caused heavy use of other methods (unicast connections) with unintended consequences and increased system load.
- Host failover is realised through multicast subscription so services are not bound to a specific endpoint.
- Failover by location. The T7 production app is hosted in separate locations where one location can fail but the essential services of the system can continue in a degraded fashion.

Future of Infiniband at DB?

Issues in Infiniband for the future at DB

- Single Vendor (NVIDIA/Mellanox)
- De-opensourcing
- Developments seems to be less intense than in prior years
- Top IB speeds are too fast hardware for Enterprise class servers.
- RDMA is useful, but the usefulness in a classic one NIC configuration maxes out at QDR speeds due to overhead of handling at the application layer. What is the point of higher IB speeds if the architecture (PCIe3.. Intel... groan) cannot support it.
- The T7 application is depending on the performance of small packets not large RDMA data streams. The "RDMA" layer is used for standard messaging and kernel bypass and not for true RDMA transfers.
- The DB is evaluating a potential future solution using Ethernet.
- However, no consistent offload technology exists on 10G and the challenge of getting high throughput from the application to the network is even more complicated.

2021 OFA Virtual Workshop

THANK YOU

Christoph Lameter, Ph.D., Senior IT Experte

Deutsche Boerse AG