X-RDMA: Effective RDMA Middleware in Large-scale Production Environments

Zhuo Song

Alibaba Operating System

Original Paper and Authors

Teng Ma, Tao Ma, Zhuo Song, Jingxuan Li, Huaixin Chang, Kang Chen, Hai Jiang, Yongwei Wu

IEEE Cluster 2019
Albuquerque, New Mexico, USA
23th Sep 2019

RDMA in Data Center

RDMA is a widely used in modern DC that offers low-latency, high-bandwidth, and server-bypassing features

RoCE is one of the most popular hardware devices that supports RDMA

Low-latency: 1-3µs
High-bandwidth: Up to 100Gb/s
Server-bypassing: Server CPU and OS aware nothing about data transfer

Recent Works: Focus on Performance

- **File System**

  Octopus [ATC' 17]
  Hdfs over Infiniband [SC' 12]

- **Key-Value Store**

  Pilaf [ATC' 13]
  Herd [SIGCOMM' 14]
  FaRM [NSDI' 14]

- **Graph Processing**

  Gram [SoCC' 15]
  WuKong(+S) [OSDI 16/SOSP' 17]

- **VM Migration**

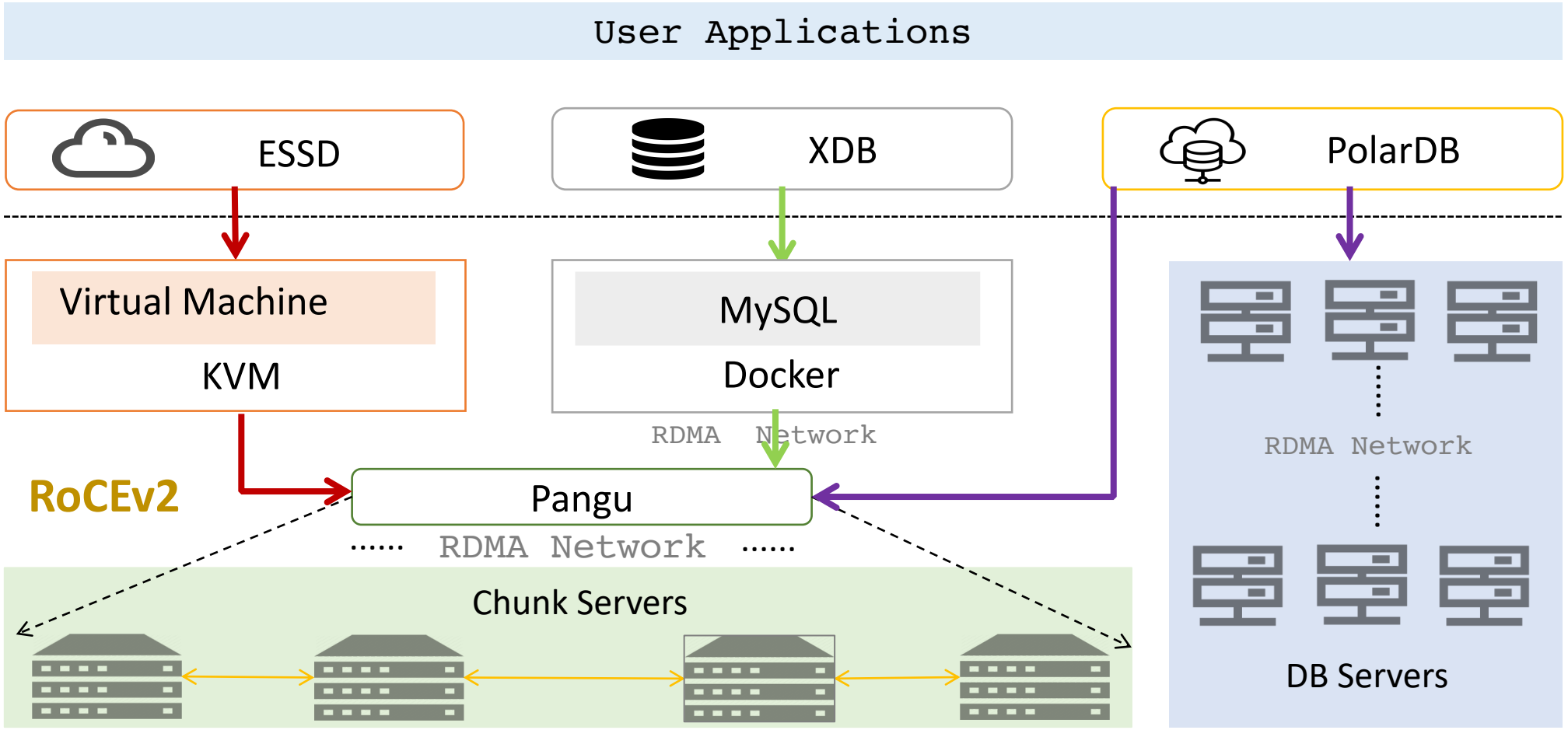
  Huang et al [Cluster' 07]

- **Deep Learning**

  AR-gRPC [HiPC' 18]

What's more?

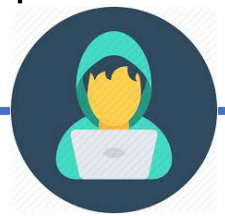
RDMA Use Cases at Alibaba



[1] A Brief History of Development of Alibaba Cloud PolarDB (https://www.alibabacloud.com/blog/)
[2] What's New in Alibaba's X-DB SQL Engine (www.percona.com/live/18/sessions/whats-new-in-alibabas-x-db-sql-engine)

Large Scale Production Issues

- Complex Programming Abstraction
  - Parameters
  - Corner Cases
  - Hidden Costs
  - Specific Implementations



At least 200 LOC

- Bugs and Performance Interferences.
  - Netstat
  - Pingmesh
  - Netfilter
  - Stress Test
  - Dynamic Tuning

Miss 10+ tools

- Conditional Performance Maximization.
  - At the same time maintain raw performance is the **major** consideration

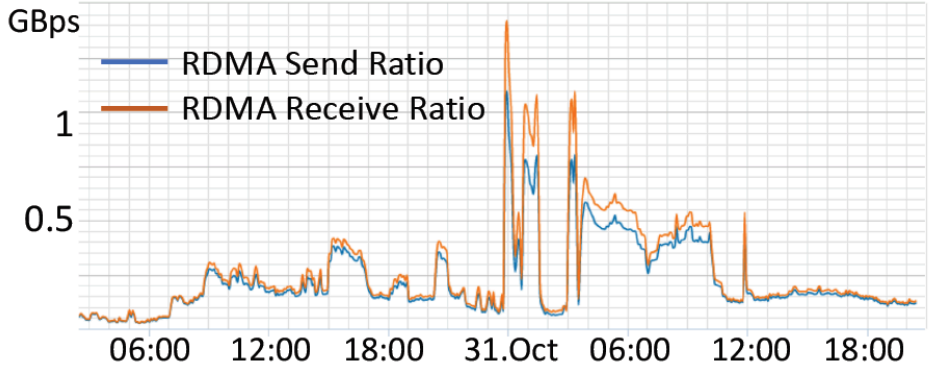
25/50/100 Gbps

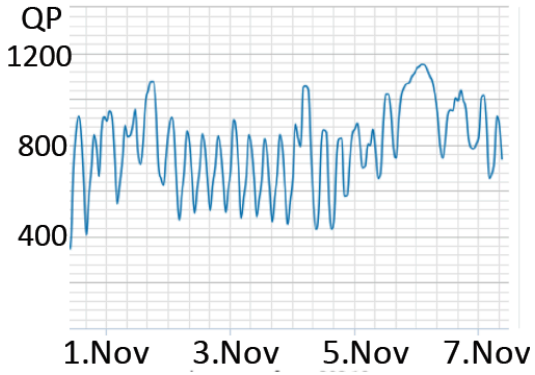
Large Scale Production Issues

Scalability Challenges.

- Issue 1: RDMA resource footprint will increase rapidly as the cluster scale.
  - E.g: full-mesh connectivity => N * M * blockserver_number * depth * message size

- Issue 2: Congestion and heavy incast exist commonly in large-scale RDMA network





- Issue 3: Slow connection establishment can harm the cluster return to steady-state.
  - the throughput of ESSD will be nearly 65% lower than the steady-state (64 machines).

Large Scale Production Issues

Lower Robustness.

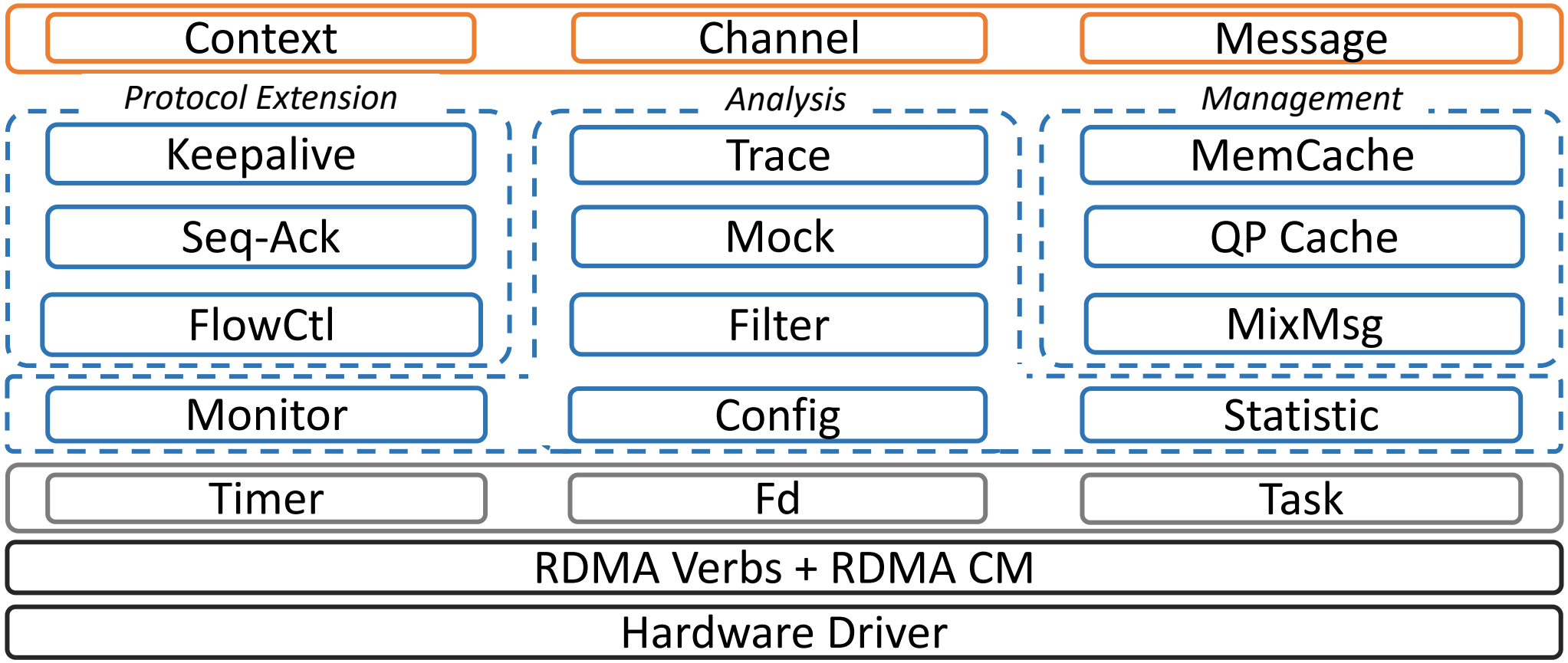
- Issue 1: the applications cannot aware the processing progress (one-sided RDMA).
  - The sender side cannot determine if the packet has been perceived by application.
  - The sender does not know the progress and keeps transmitting continuously
  - A receiver-notready (RNR) error will be raised when there is no buffer available.

- Issue 2: Native RDMA library and RNIC cannot ensure the peer is active all the time.
  - The chunk servers has been disconnected, resources are held until future communication
  - Some servers will still occupy connection resources (nearly at GB scale in Pangu)
  - Solved by TCP keepalive option

X-RDMA Design

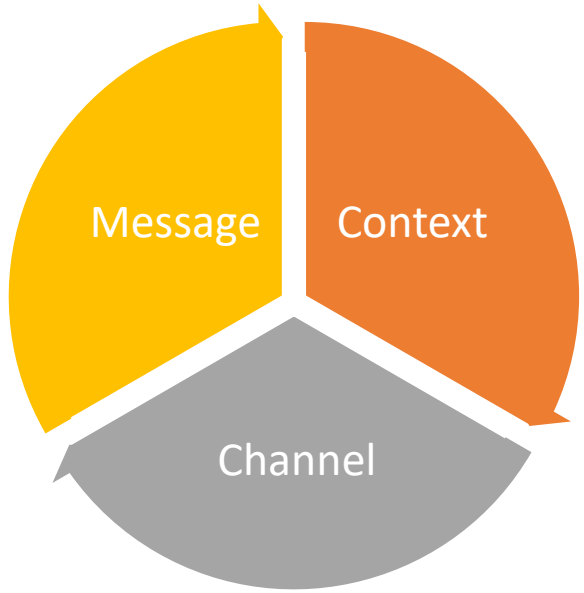
- Architecture
- Thread Model
- Message Model
- Resource Mangement

Overall Architecture



Overall Architecture (Con.)

APIs	Descriptions
send_msg	common routine of sending message to remote
polling	polling the context to check events/messages.
get_event_fd	get the xrdma fd to do select/poll/epoll
(de)reg_mem	register/deregister RDMA-enabled memory
set_flag	dynamic changing configurations
process_event	handle event notified by fd
trace_request	trace information of the request message



Thread Model

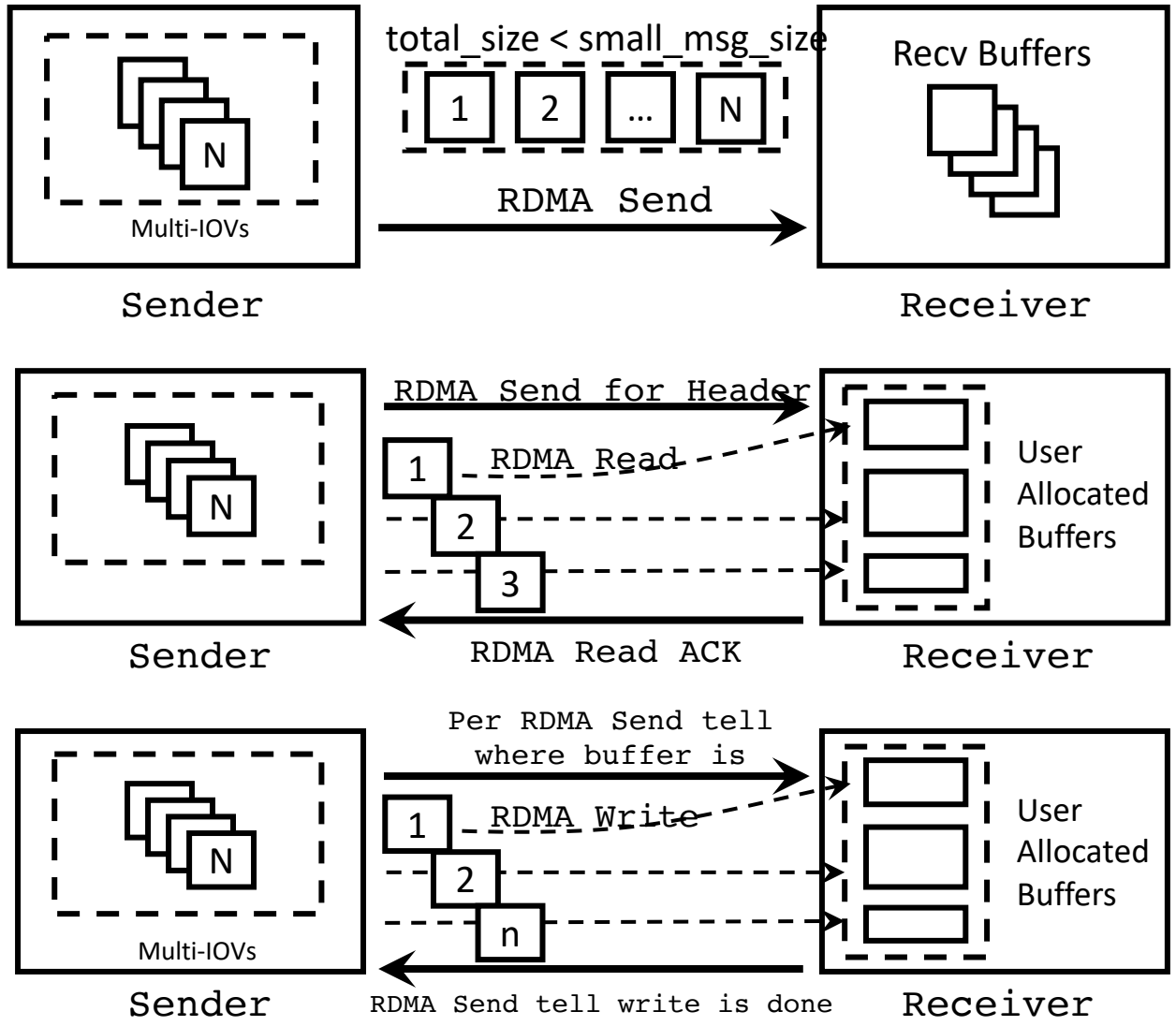
Lower Latency & Lower Resource Footprint.

- Lock-free, Atomic-free, No-syscall.
  - Avoids lock + only allows atomic operations / syscall on **non-critical** paths
  - Reduce the overheads in **bus locking** and **context switching**

- Run-to-complete
  - Context, Channel, Mem Cache, QP cache, etc only initialized once.

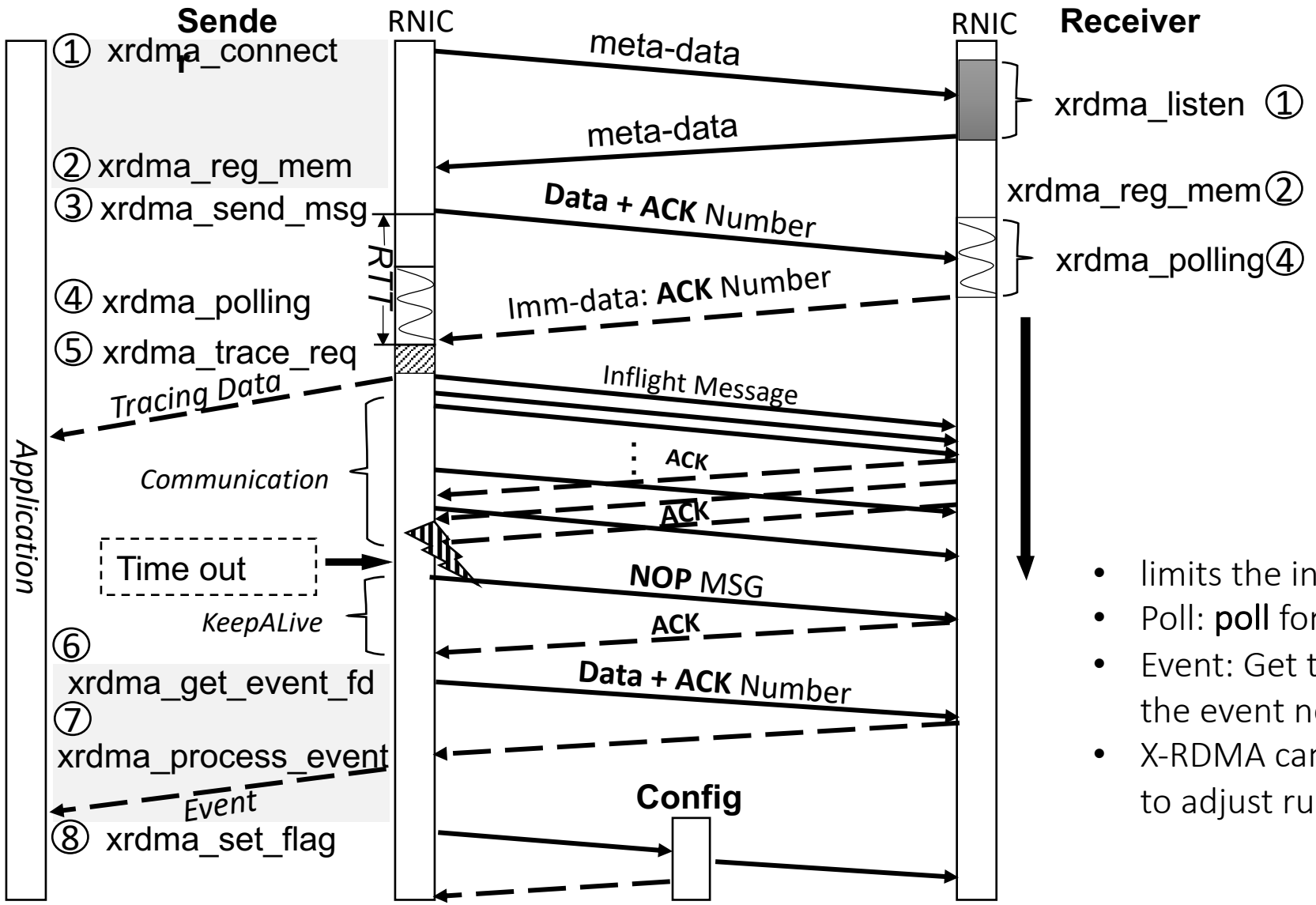
- Hybrid-polling.
  - Switch between **Epoll** and **Busy Polling**.
  - Per-thread **Timer** for monitoring, protocol, etc

Message Model



- Small Message
  - RDMA Send/Recv + Reserved Bounced Buffer
- Large Message
  - RDMA Write/Read + Prepared Buffer
- Built-in RPC
  - RDMA Read (fetch back the response)
  - To remedy heavy out-bound operations

Work Flow



- limits the inflight messages as **depth** (< CQ depth)
- Poll: **poll** for the completion (4) of the message
- Event: Get the event **fd** (6) first and then handle the event notifications (7)
- X-RDMA can change **configuration** dynamically (8) to adjust running state

Resource Management

Reduce Memory Footprint & shorten establishment time.

- Memory Cache.
  - Contain multiple **MR**, automatically or manually shrunk/extend capacity
  - Set each MR to 4MB to avoid performance downgrading (4KB in **LITE [SOSP' 17]**)

- QP Cache
  - Per-Thread
  - Setting it as **`IBV_QPS_RESET`** status
  - Release to QP Cache
  - Re-use QP from QP Cache

PROTOCOL EXTENSIONS

- KeepAlive
- Seq-Ack Mechanism
- Flow Control

KeepAlive

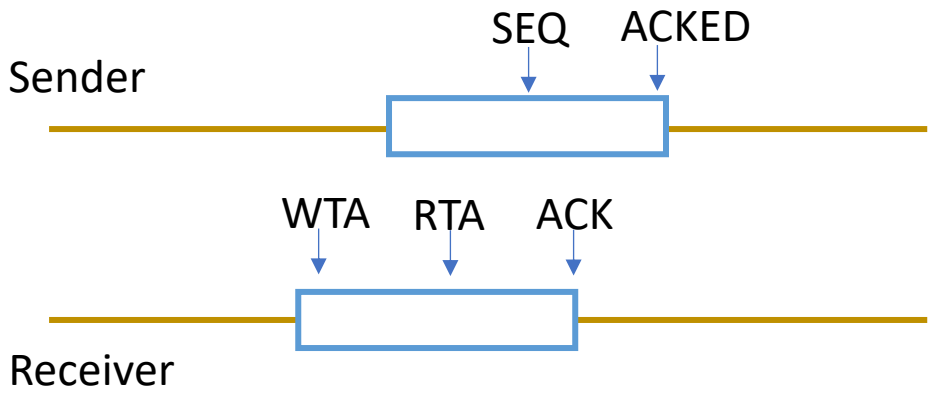
Handle Connection Leak and Improve Robustness.

- Memory Leak.
  - Timeout, Network fault, Machine crashing
- Request Probe.
  - RDMA Write, payload is **zero** size (zero message)
  - Triggered condition (fails to communicate with peer side > **S** ms)
  - Resource (e.g., QP) should be released immediately to avoid connection leaks.

Seq-Ack Window

Sender

```
2:  procedure SEND_MESSAGE(msg)
3:      QP.tx.seq++
4:  procedure RECV_MESSAGE(msg)
5:      for i in range(QP.tx.acked to msg.ack) do
6:          call_on_acked(messages[i])
7:      QP.tx.acked = msg.ack
8:  procedure TIME_OUT(timer)
9:      if deadlock ocurred then
10:         send_message(NOP_MSG)
```



Receiver

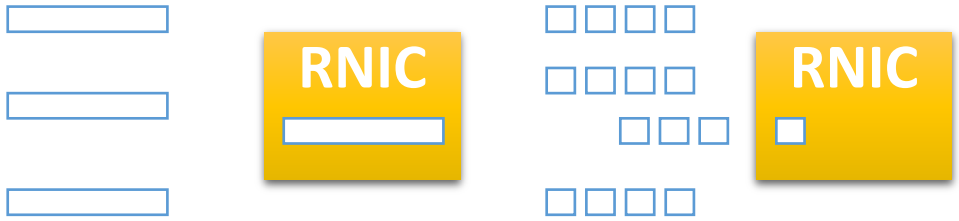
```
11: procedure SEND_MESSAGE(msg)
12:     QP.rx.wta++
13:     if need_rdma_read(msg) then
14:         do_rdma_read(msg)
15:     else
16:         msg.recved = true;
17: procedure RECV_MESSAGE(msg)
18:     QP.rx.acked = QP.rx.rta
19:     msg.acked = QP.rx.acked
20: procedure RDMA_READ_DONE(msg)
21:     msg.recved = true
22:     if msg.id == QP.rx.rta then
23:         QP.rx.rta++
24:         while QP.rx.rta < QP.rx.wta & msgs[QP.rx.rta].recved
    do
25:             QP.rx.rta++
```

- ACK - current received sequence number;
- SEQ - current sending sequence number;
- ACKED - current acknowledgment number sending to receiver;
- RTA - current acknowledgment number which is ready to ack;
- WTA - current acknowledgment number which is wait to ack;

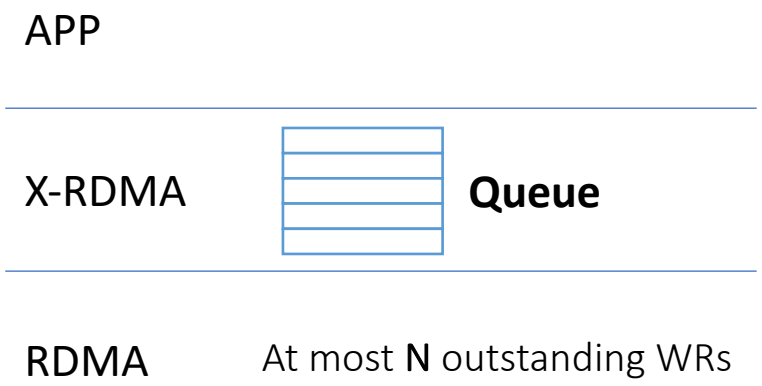
Flow Control

- DCQCN [SIGCOMM' 15] perform heavy incast in the large-scale cluster.
  - DCQCN is a passive control, incur harmful effects before the reaction works
  - More CNP and PFC pause frames are generated due to the heavy incast

- Fragmentation.
  - Large-size request block RNIC



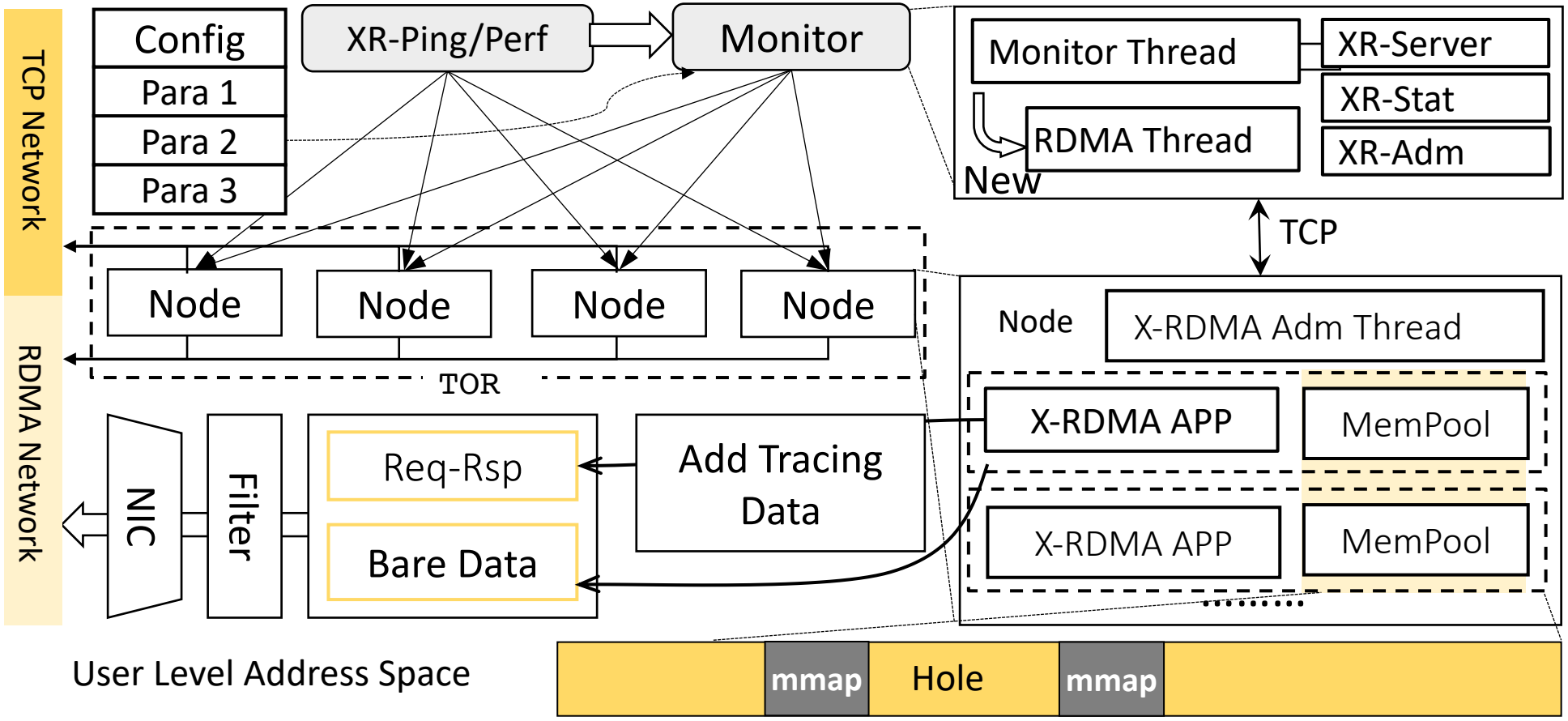
- Queuing.
  - Network congestion



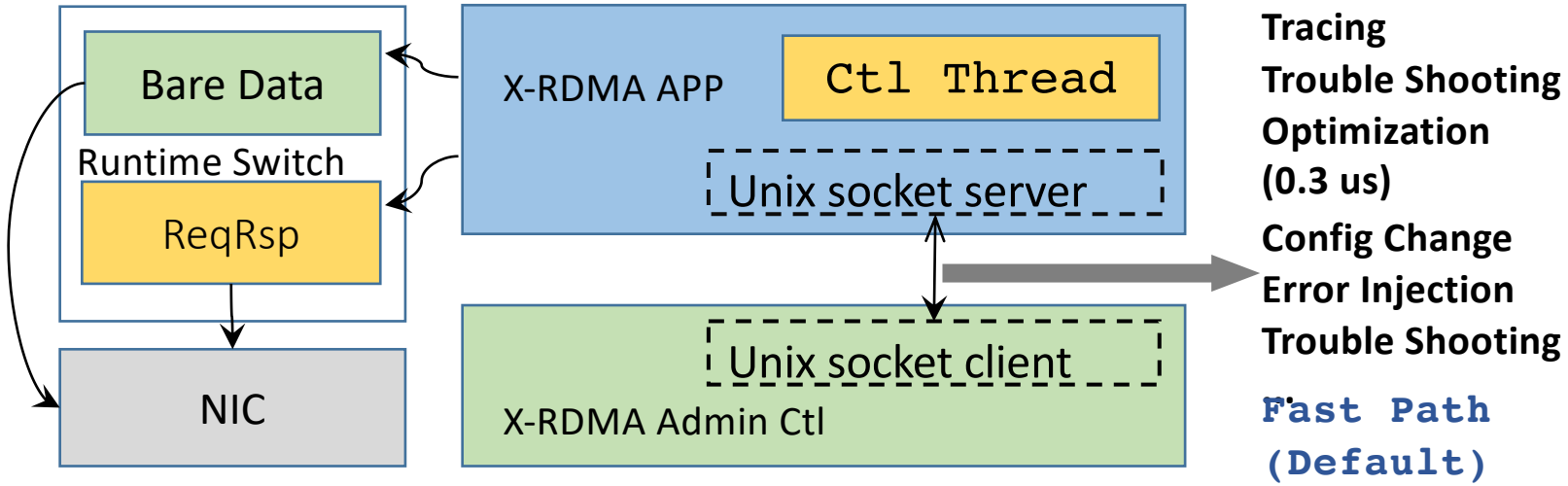
ANALYSIS FRAMEWORK

- Tracing
- Monitoring
- Extra Schemes

Overview


TCP Network
RDMA Network
Config
Para 1
Para 2
Para 3
XR-Ping/Perf
Monitor
Monitor Thread
XR-Server
XR-Stat
XR-Adm
RDMA Thread
New
TCP
Node
Node
Node
Node
TOR
Node
X-RDMA Adm Thread
X-RDMA APP
MemPool
X-RDMA APP
MemPool
NIC
Filter
Req-Rsp
Bare Data
Add Tracing Data
User Level Address Space
mmap
Hole
mmap

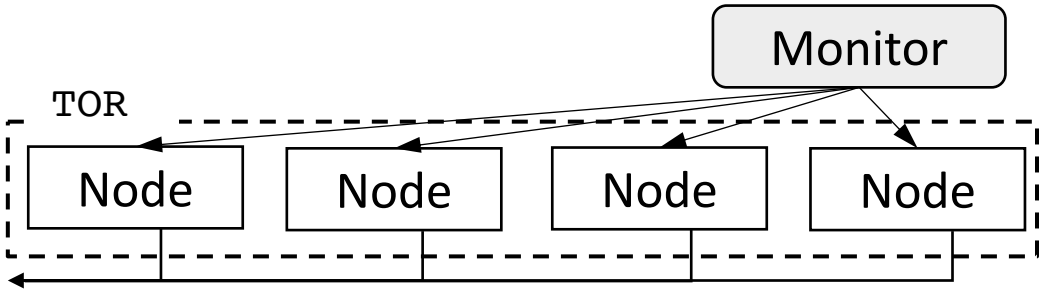
Tracing



- Three Case-by-Case Methods.
  - **Long Request Latency**: $T_2 - T_1 - T_{off}$ (clock synchronization service)
  - **Slow Polling**: Count the time interval between two polling operation
  - **Performance Bottleneck:** Record the execution time of critical code segments

Monitoring

- XR-Stat.
  - **Per-connection statistics**: PFC status, Queue Drop Counter, Buffer Utilization
- XR-Ping
  - **Full-mesh connection status**: Ping all machines in the ToR layer.

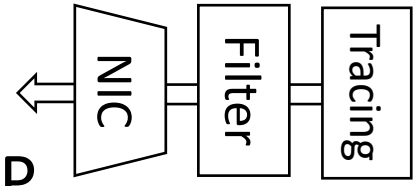


- XR-Perf
  - **Stress Test**: e.g., elephant and mice flow [KBNet' 17]

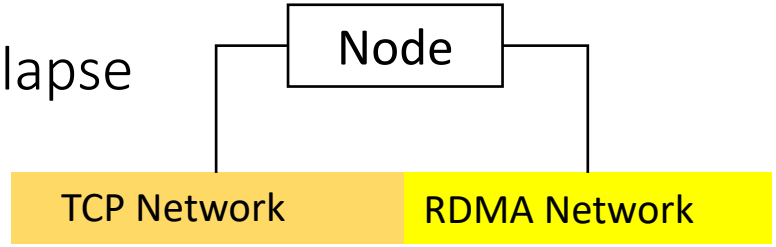
Extra Schemes

- Memory Cache Isolation
  - memory cache will be assigned to a higher address space via mmap
  - marked to avoid conflict with other threads' addresses

- Emulate Fault (Filter)
  - Enable/Disable Dynamically (detecting dropped messages, slow messages)



- Switch between RDMA and TCP
  - heavy congestion, high-degree incast, protocol stack collapse



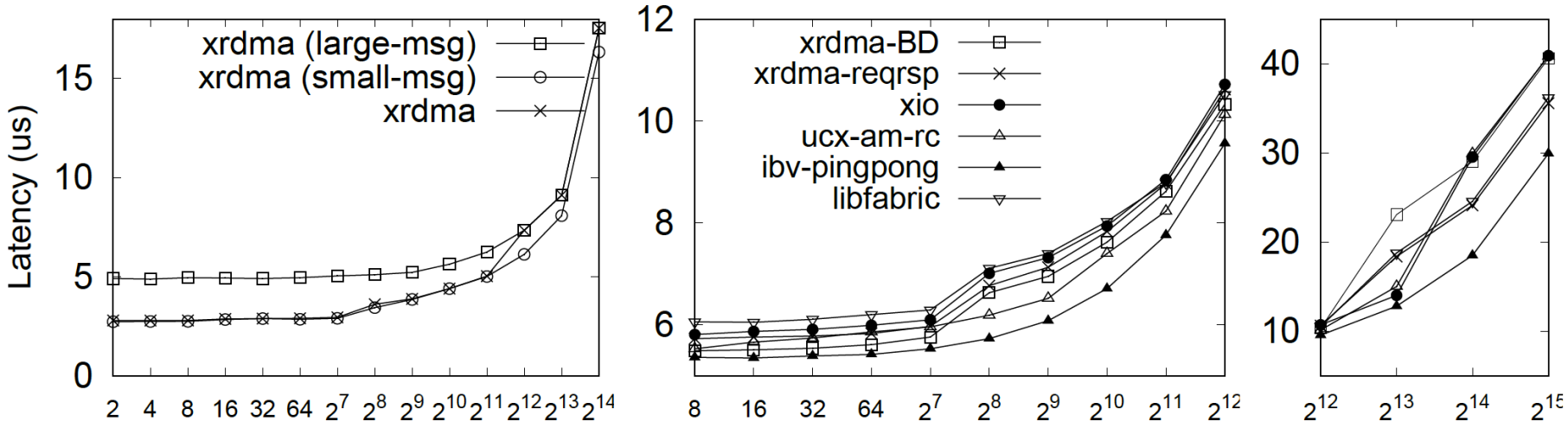
EVALUATIONS AND EXPERIENCE

Deployment

- Protocol: RoCEv2
- Dual-port 25Gbps Mellanox ConnectX4-Lx RNIC
- Applications: X-DB, ESSD, Pangu, and PolarDB [VLDB' 18]
- Real-world workloads during shopping transactions

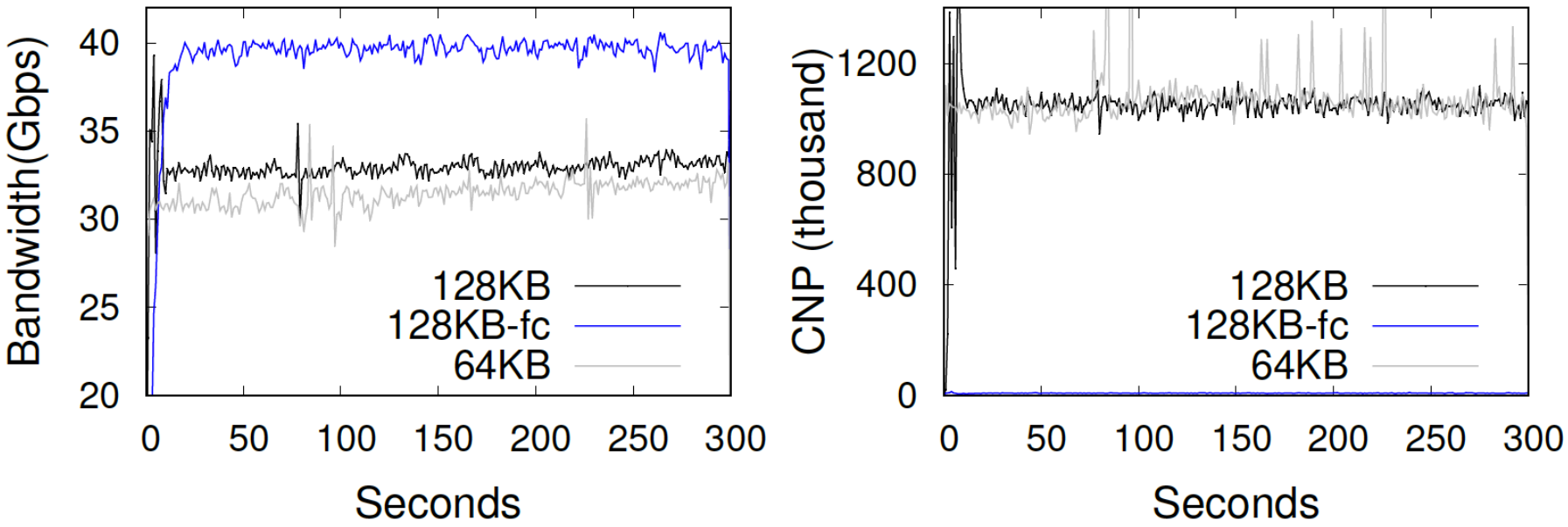
At Alibaba, over 4000 servers are deployed with X-RDMA using RoCEv2 protocol.

Performance Overview



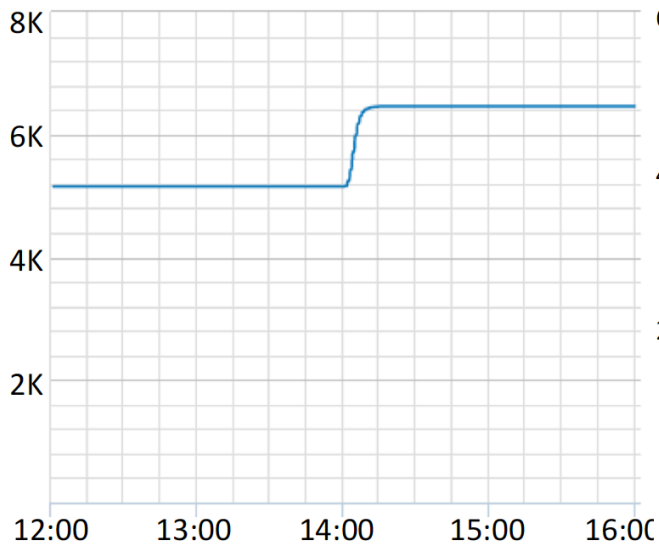
X-RDMA performs 5%/10% lower latency (5.60us) than ucx-am-rc of UCX (5.87us) and Libfabric (6.20s).

Robustness Enhancement

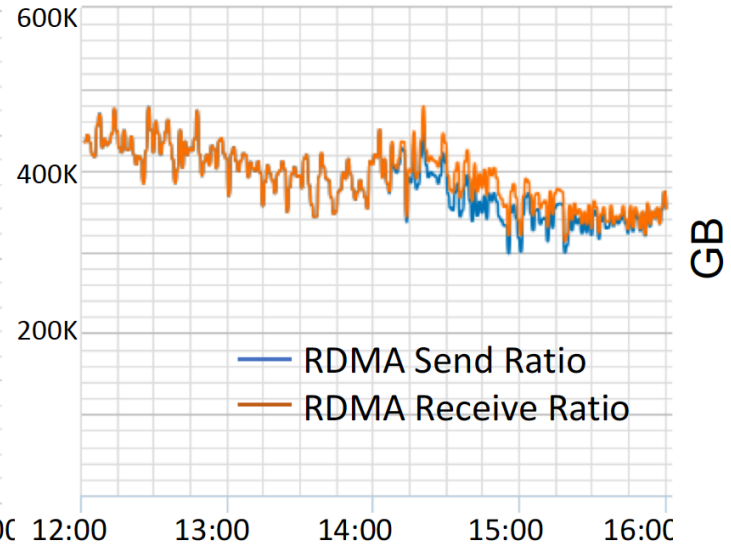


The bandwidth can be improved by around 24%, the average CNP number is reduced to 1~2%

Robustness Enhancement (Con.)



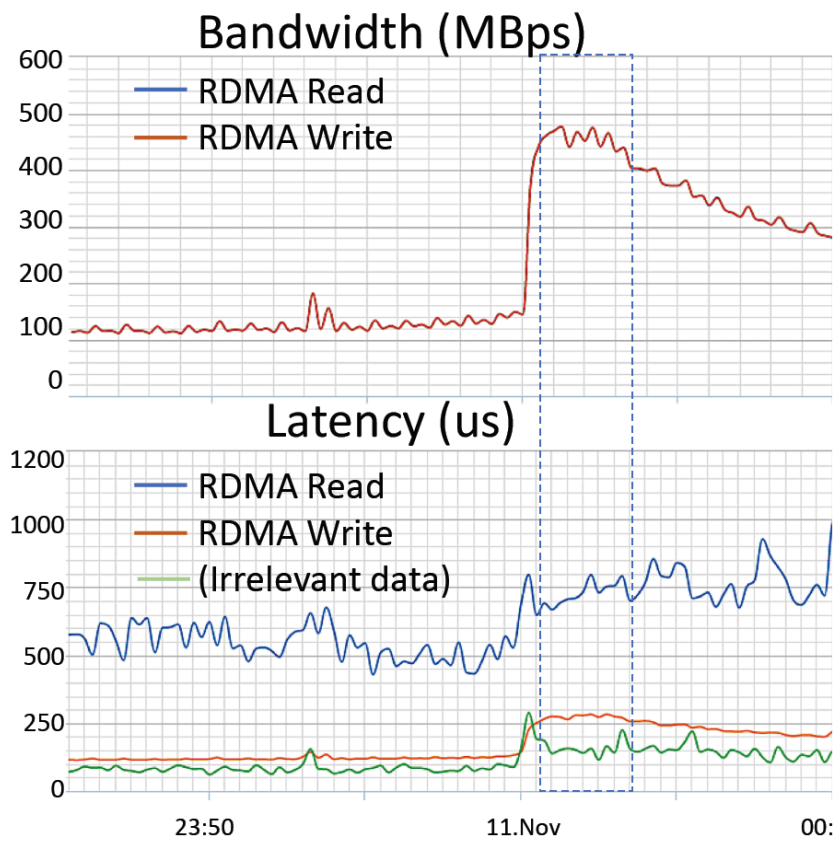
(a) Pangu QP Num

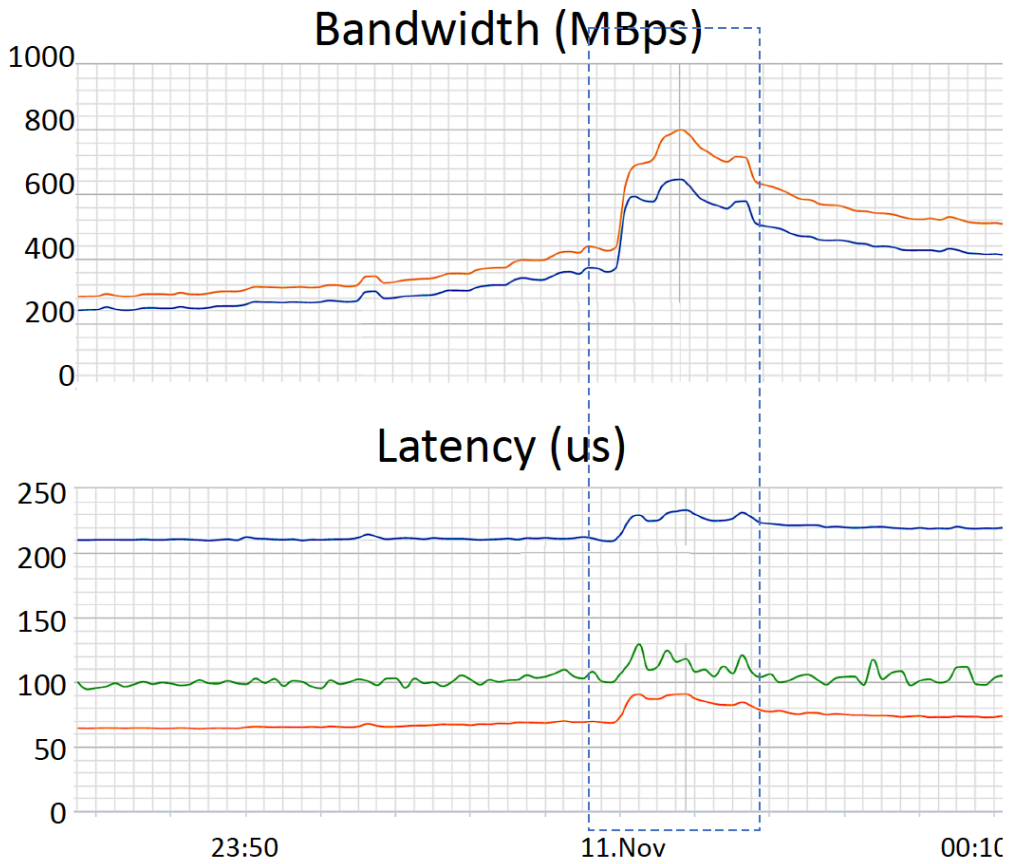


(b) Pangu IOPS

Online upgrading will increase the QP number rapidly but will not harm the performance or result in jitter.

Production Environment





Jitter is mitigated effectively under pressure

Extra Experience

- Influence of RNIC cache is limited.
  - influence on performance is below 10%
- Shared Receive Queue (SRQ)
  - SRQ can potentially cause network jitter (is not recommended under 10K connections).
- Physical Continuous Memory
  - Non-continuous page still has comparable performance and less fragmentations (non-continuous, continuous, hugepage)

More experience and evaluations are shown in our paper

Cooperation is welcome!

Coming work…

- Our team is working on many research directions in system software including but not limited to OS performance optimization, system debugging and tracing, networking optimization and diagnose, scheduling and resource optimization, hypervisor …
- As the infrastructure, our scenarios cover cloud native, serverless, DC networking, distributed cloud storage …
- We kickoff system research projects just in recent one year and all those research directions above are long-term.
- Potential top papers are coming soon or waiting for you to develop ☺

Contact:

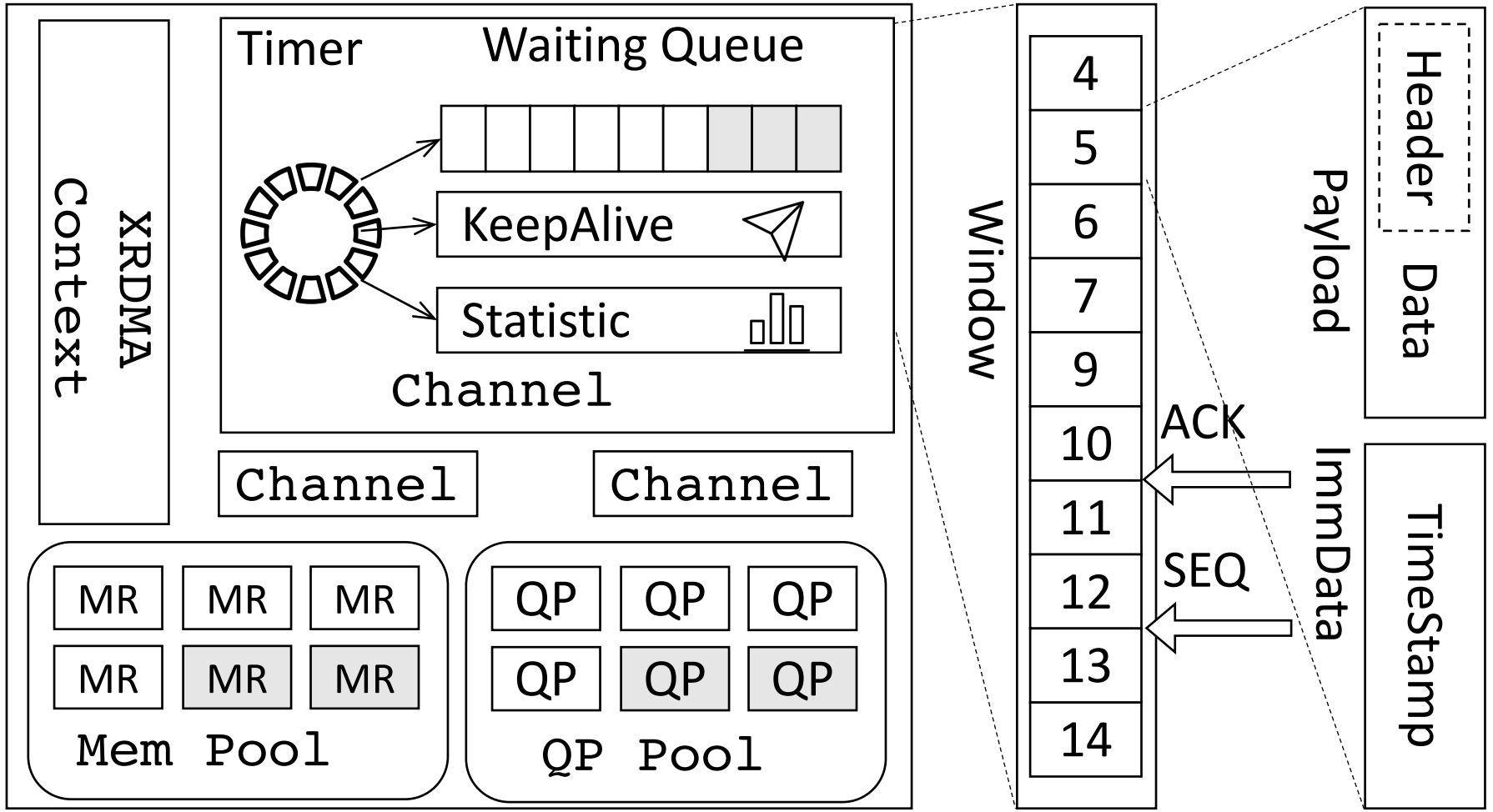
宋卓（文侑）13581906251 （钉钉/微信）
songzhuo.sz@alibaba-inc.com

Thank You

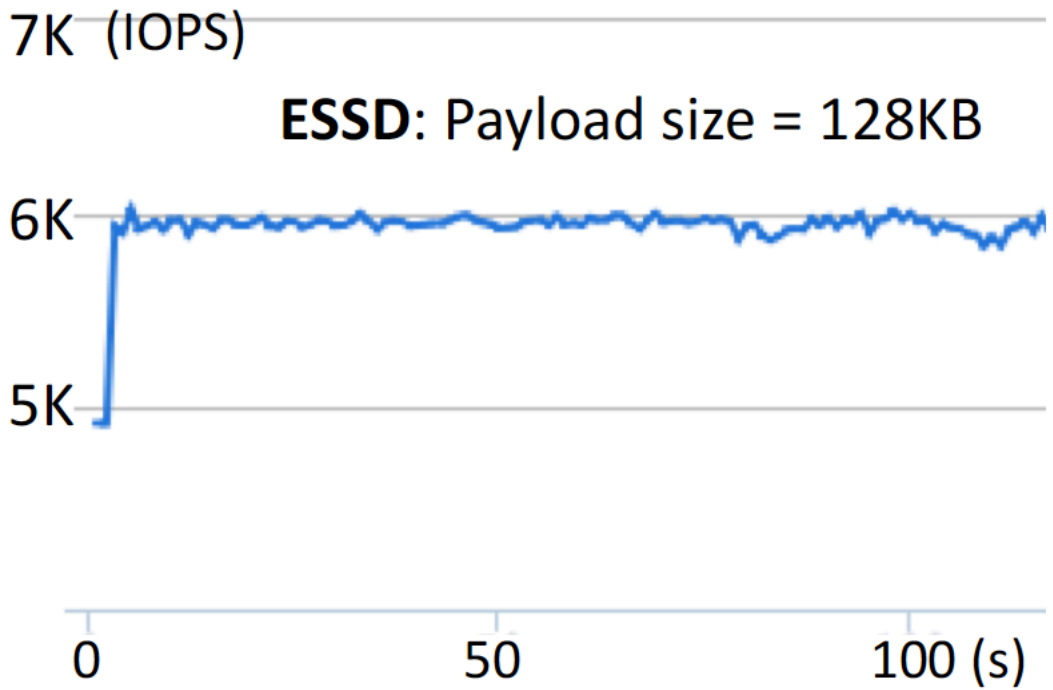
Q&A

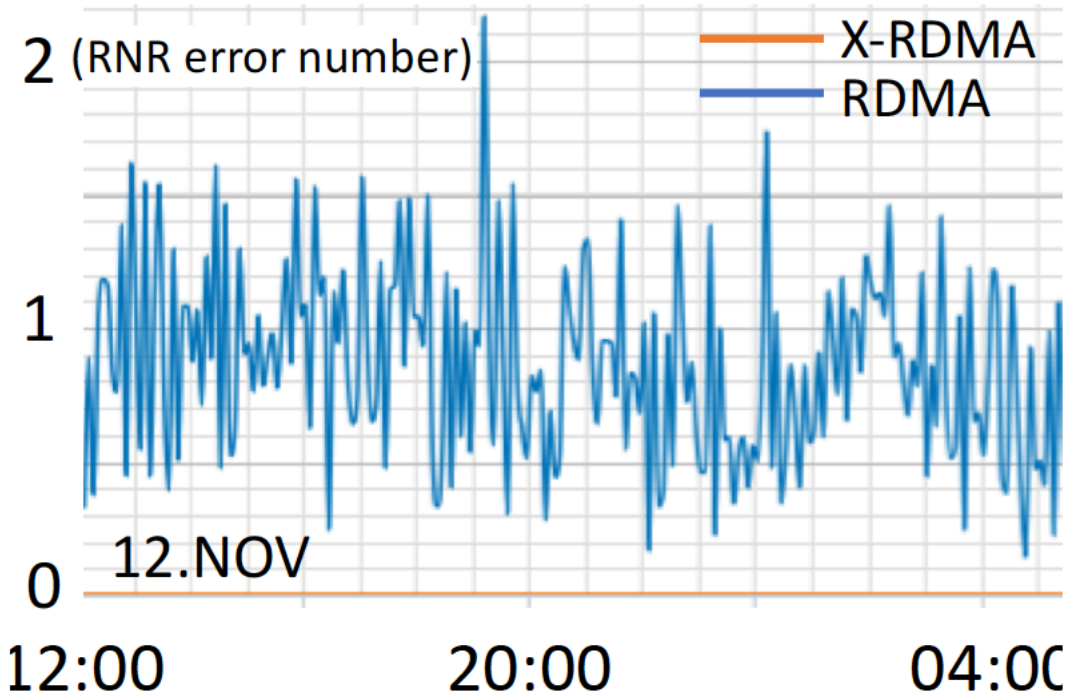
Back-up

Arch


XRDMA Context
Timer
Waiting Queue
KeepAlive
Statistic
Channel
Channel
Channel
MR
MR
MR
MR
MR
MR
Mem Pool
QP
QP
QP
QP
QP
QP
QP Pool
Window
4
5
6
7
9
10
11
12
13
14
ACK
SEQ
Payload
Header
Data
ImmData
TimeStamp

Aggregate IOPS & RNR Free

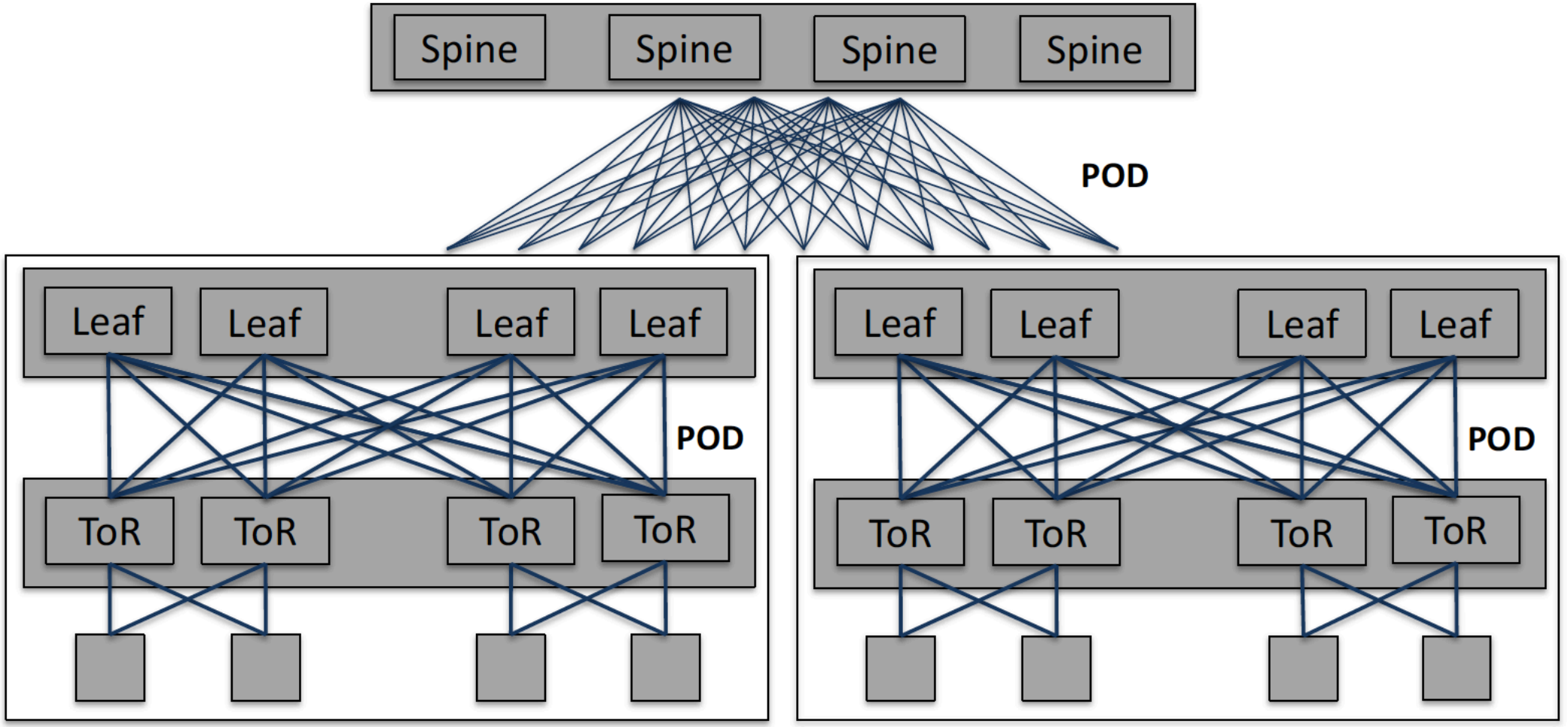




Debugging

Bug Type	Tracking Method
heavy Incast	tracing, XR-Stat
broken network	keepAlive, XR-Ping
jitter	tracing, XR-perf
long tail	tracing, XR-perf
bugs hard to reproduce	filter
memory leak or crash	isolated memory cache

Topo


Spine
Spine
Spine
Spine
POD
Leaf
Leaf
Leaf
Leaf
POD
ToR
ToR
ToR
ToR
Leaf
Leaf
Leaf
Leaf
POD
ToR
ToR
ToR
ToR