Alibaba Cloud

SMC-ERM: A fast remote memory communication method based on SMC socket

Dust Li/D. Wythe

dust.li@linux.alibaba.com/alibuda@linux.alibaba.com

2024/09/18

Alibaba Cloud

Agenda

01. Problems statement

02. Design

03. Preliminary results

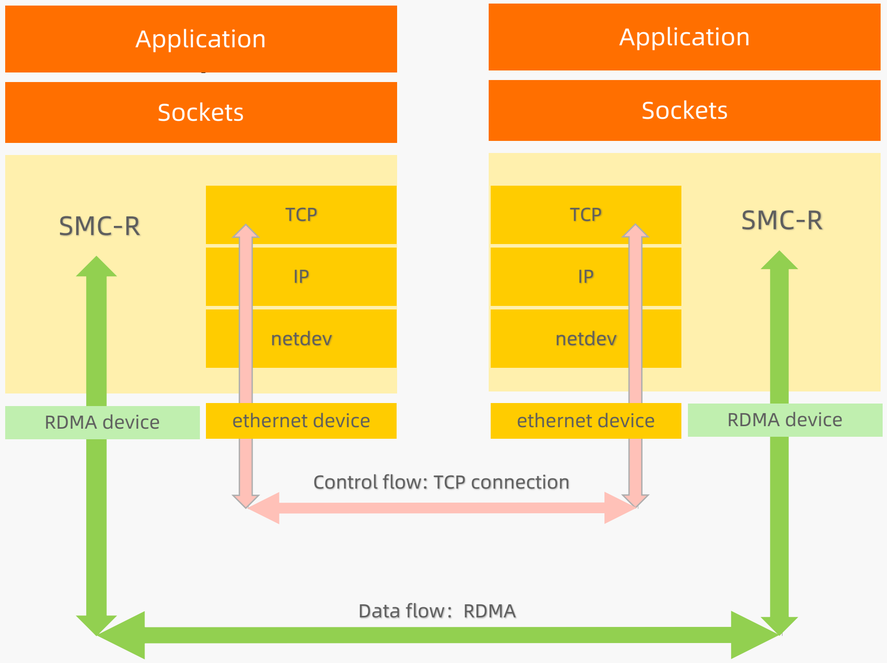
04. Status + Future work

05. Q & A

Alibaba Cloud

01. Problems statement

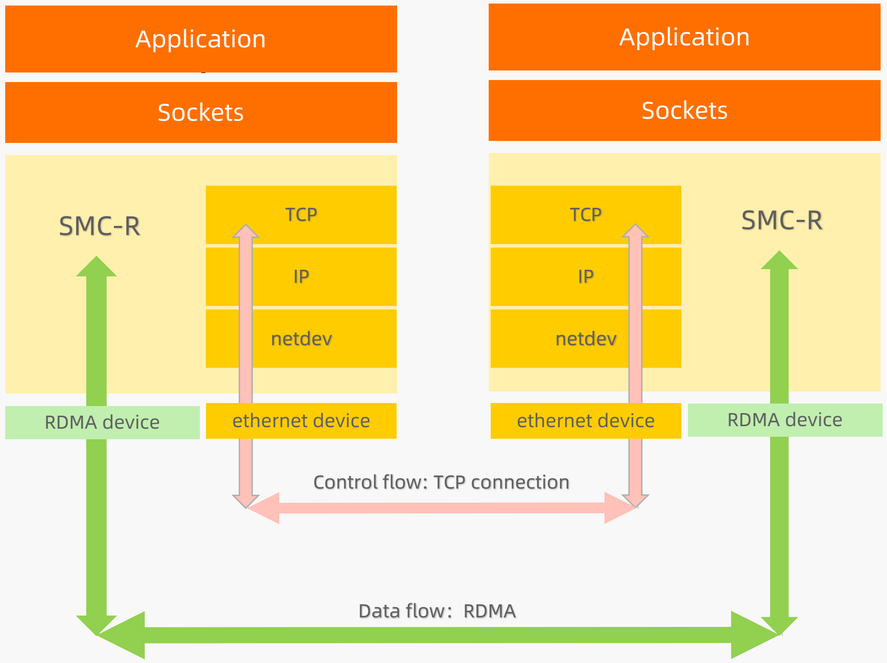
What is SMC-R ?



SMC is an effort to boost the performance of TCP applications in datacenter without any code change

1. Provide compatible APIs with TCP socket
2. use RDMA for data transmission at the lower level.
3. Implement in the kernel, works with static linked apps. Like golang

Ways to use SMC-R



1. using AF_SMC or IPPROTO_SMC explicitly in application

   socket(AF_INET, SOCK_STREAM, 0) →

   socket(AF_SMC, SOCK_STREAM, 0)

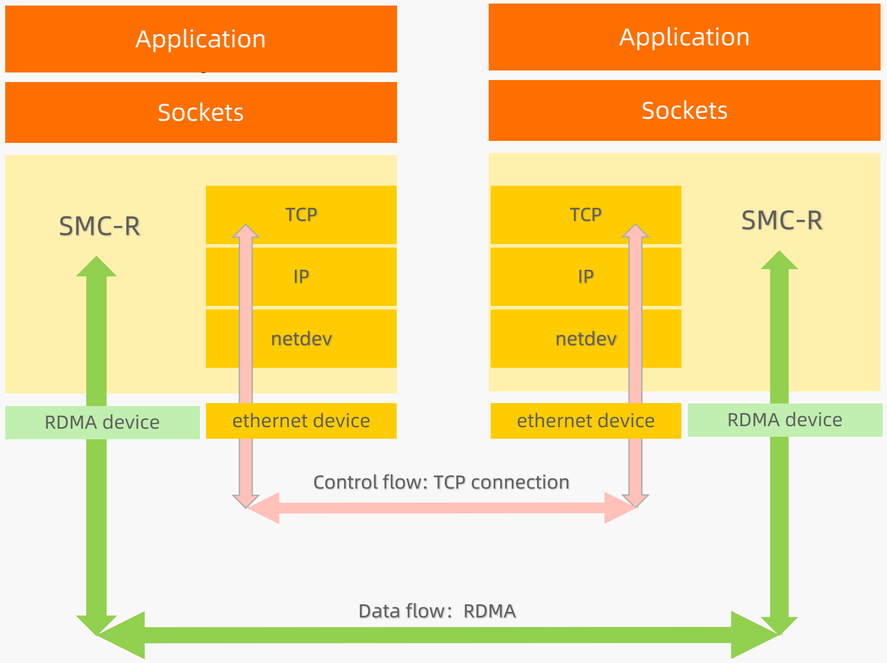
   socket(AF_INET, SOCK_STREAM, IPPROTO_SMC)

2. LD_PRELOAD: add a smc_run prefix before application, *smc_run netperf*

3. eBPF dynamic replacement: same like mptcp

   - filter by addr/port/Process Name etc.

Benefits & Drawbacks



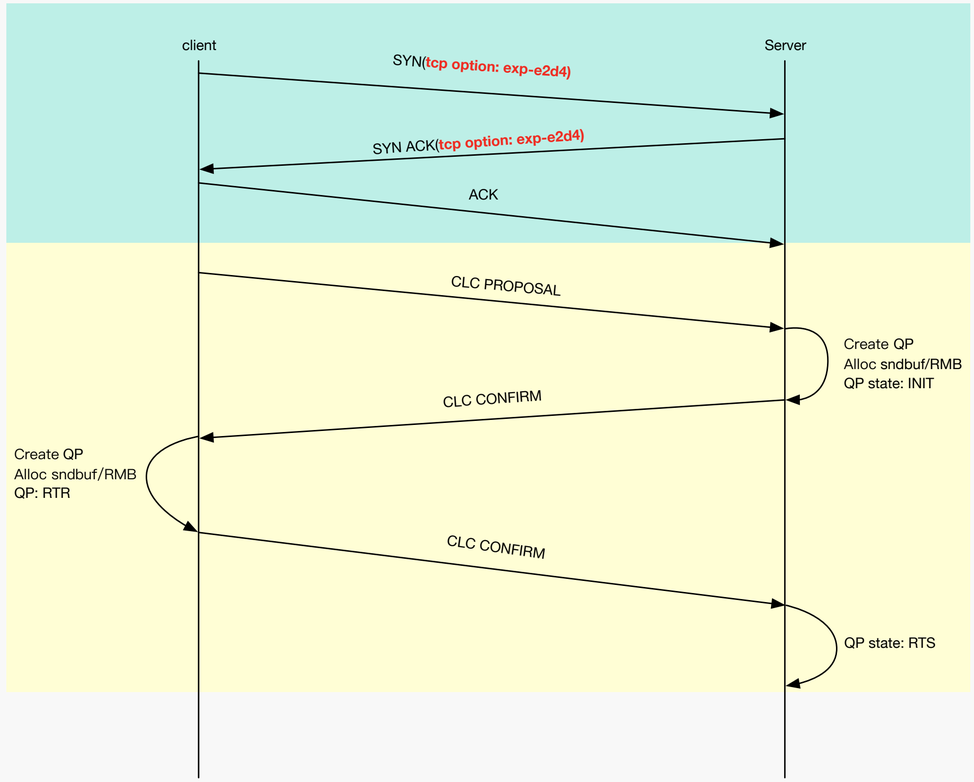
benefits

- Network performance can be improved without any code change for many applications. Like redis/Kafka

drawbacks

- Doesn’t support zerocopy currently

SMC-R Control path

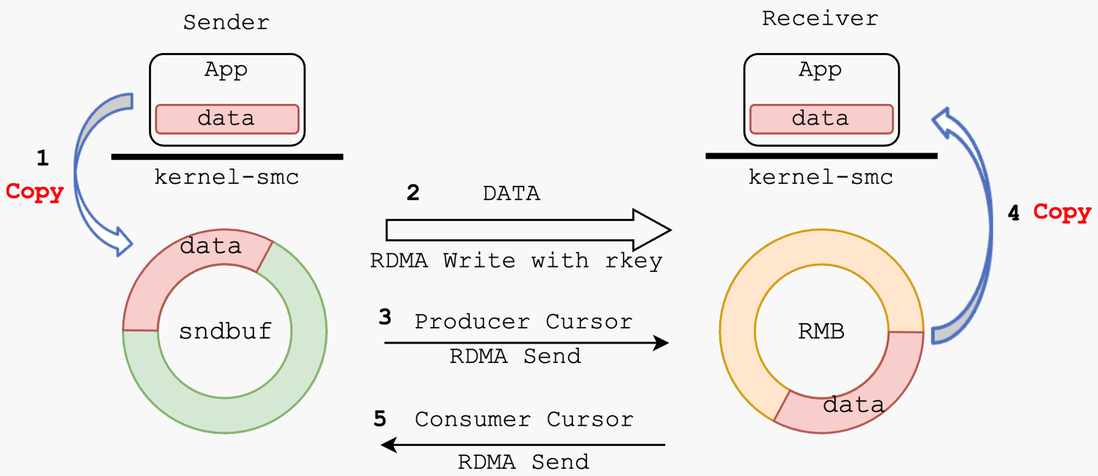


1. A TCP experimental option to indicate if both side support SMC

2. An extra SMC handshake after TCP 3-way handshake to prepare SMC connection resources

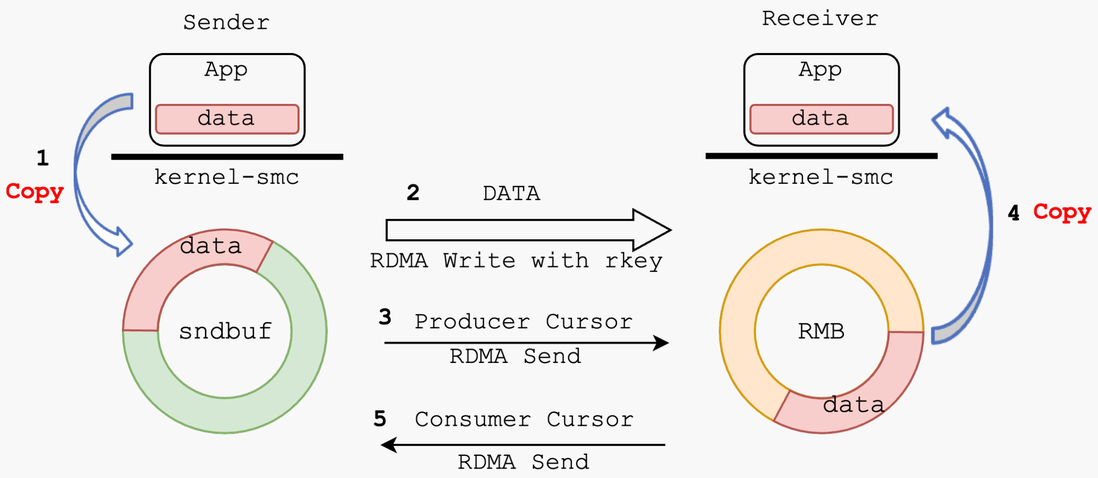
3. Fallback to the TCP connection if anything goes wrong

SMC-R Data path



1. Data is copied to SMC kernel sendbuf with sendmsg() syscall
2. Data is moved by RDMA device using RDMA Write
3. Sender notifies the receiver to update its cursor
4. Data is copied to user space from RMB(Receive Memory Buffer) with recvmsg() syscall
5. Receiver notifies the sender to update its sndbuf cursor

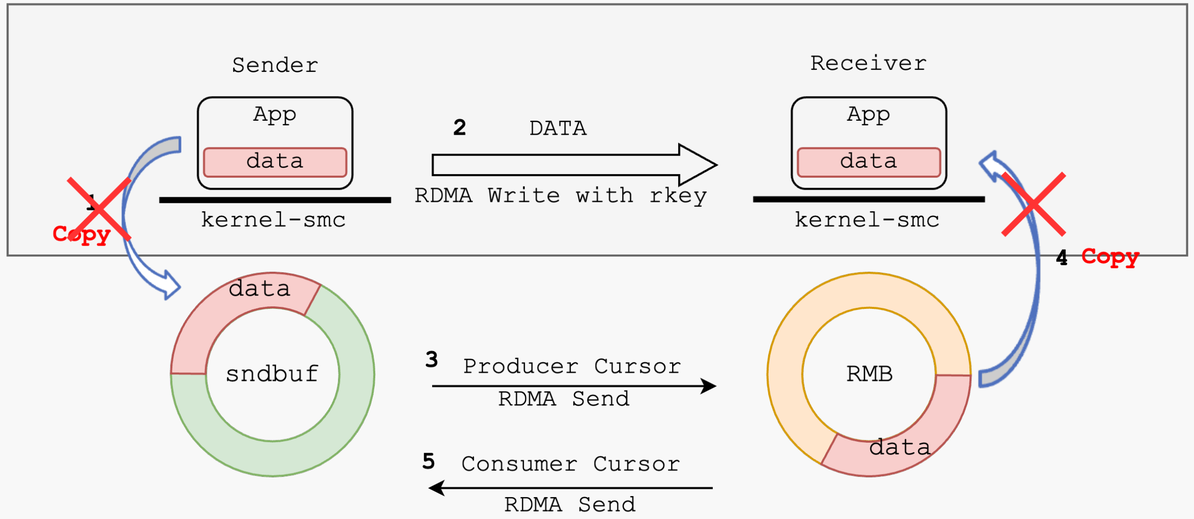
Problems with SMC-R Data path



1. Data is copied to SMC kernel sendbuf with sendmsg() syscall
2. Data is moved by RDMA device using RDMA Write
3. Sender notifies the receiver to update its cursor
4. Data is copied to user space from RMB(Receive Memory Buffer) with recvmsg() syscall
5. Receiver notifies the sender to update its sndbuf cursor

- For 1 data transfer, 2 CPU memory copies, 1 RDMA copy
- Memory copies limit the throughput

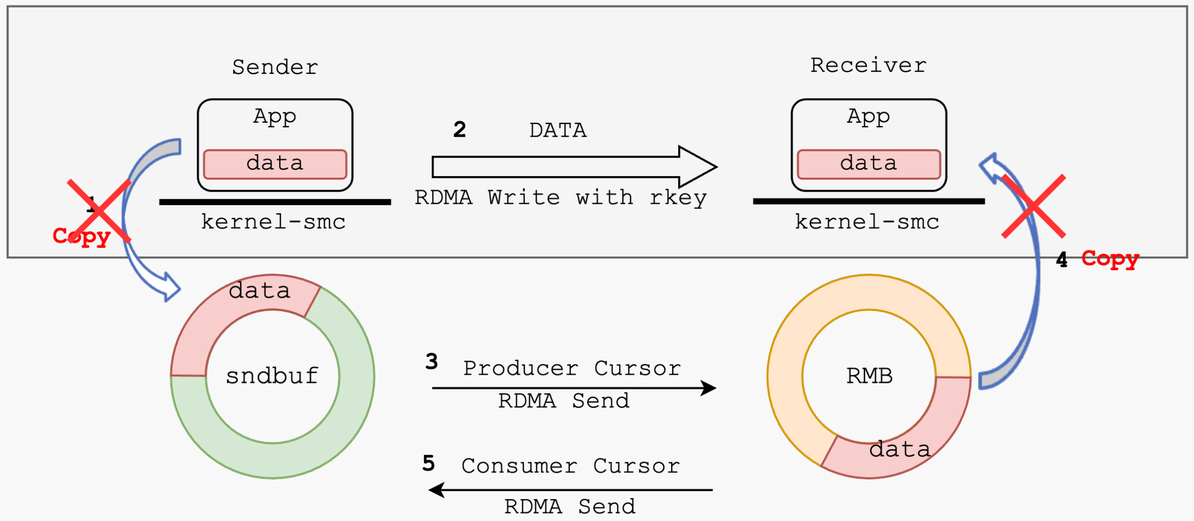
SMC-R zerocopy



- Communication is all about moving memory
- Direct moving data from Sender application to Receiver application

- SMC-ERM(Extended Remote Memory): Direct memory access on top of SMC.

Why Extend APIs ?



Can we reuse the existing Zerocopy APIs ?

ret = send(fd, buf, sizeof(buf), MSG_ZEROCOPY);

res = getsockopt(fd, IPPROTO_TCP, TCP_ZEROCOPY_RECEIVE, &zc, &zc_len);

Difference with TCP zerocopy:

1. Memory Pin is necessary for RDMA
2. TCP tx & rx zerocopy are seperated

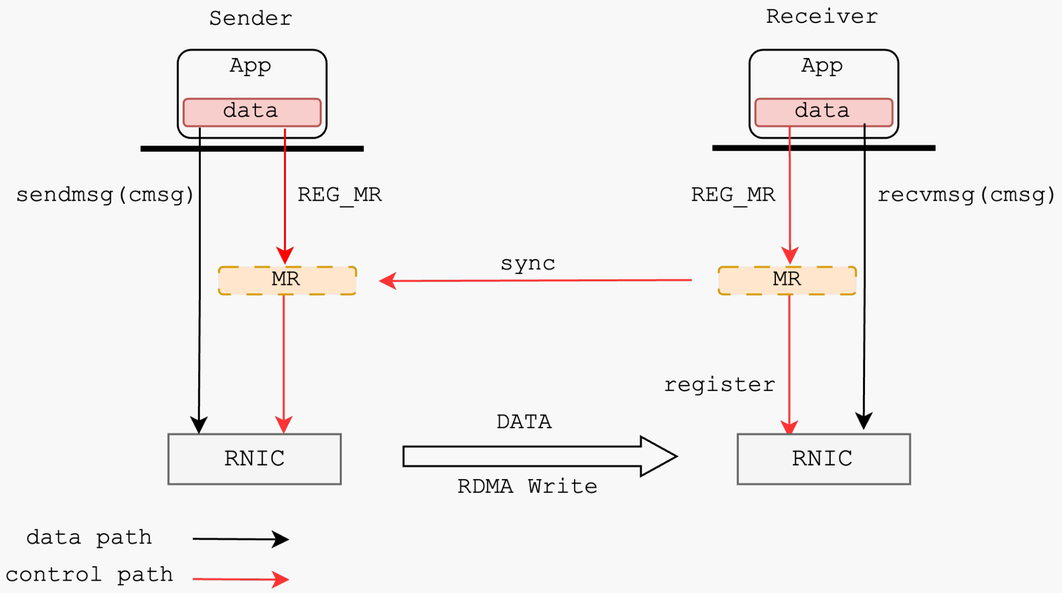
For RDMA, it's easy to support zero copy:

1. Received data goes to specific RC QP directly with no header
2. RDMA read/write can access any registered memory, both local and remote

Alibaba Cloud

02. Design

SMC-ERM Design



Goal

1.High Performance: Competitive performance over user space RDMA

2.Easy to use: simple datapath APIs

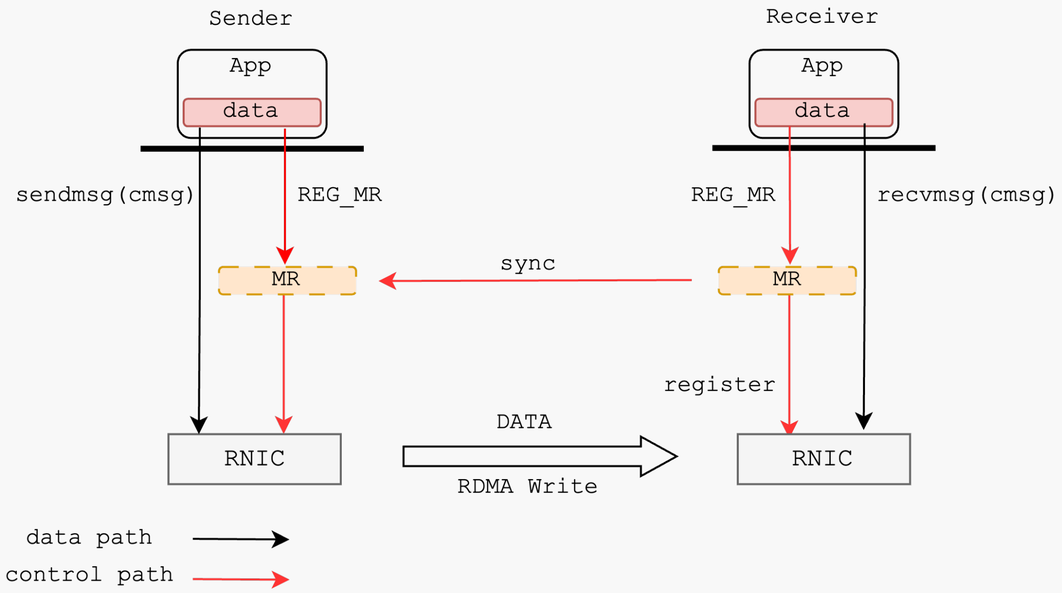
How it works

1.user memory is directly registered to RDMA device

2.provide memory directly accessible to the peer socket

3.Meta data is stored in cmsg, user call sendmsg()/recvmsg()

SMC-ERM Design



Two set of extended APIs

- control path: setsockopt(REG_MR/DEREG_MR)
- data path : sendmsg(cmsg)/recvmsg(cmsg)

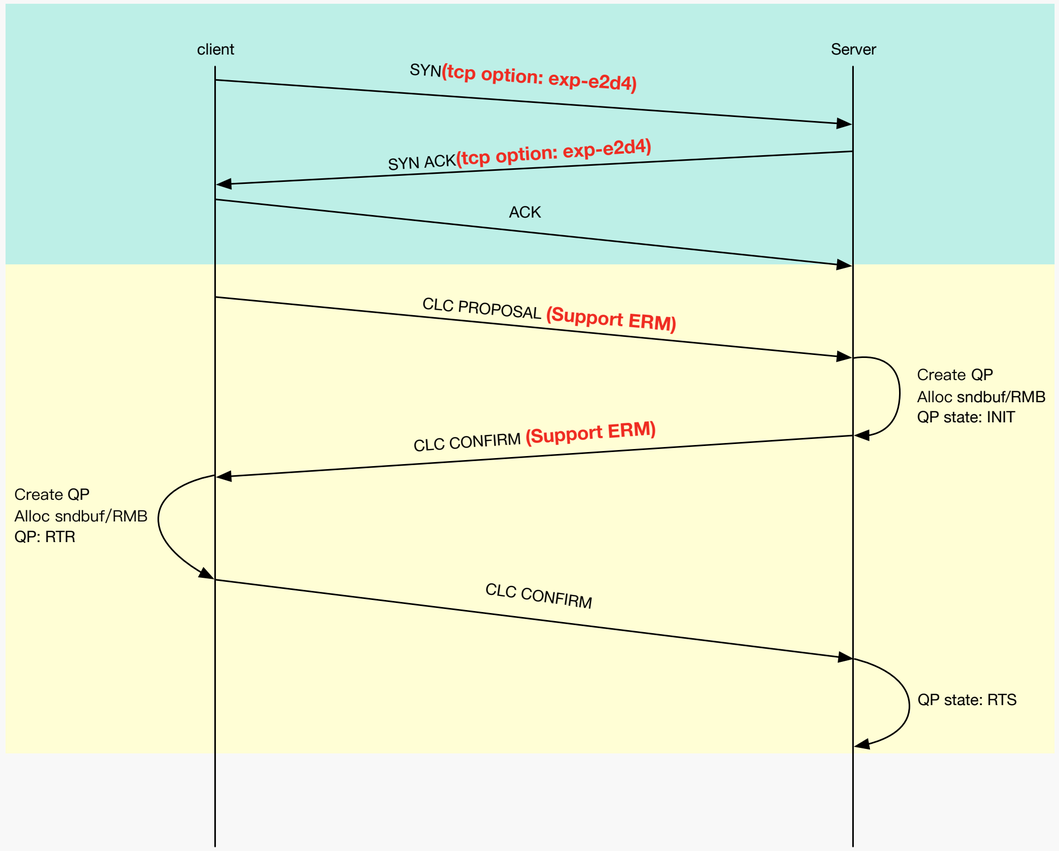
Control path

- handshake
- memory register & unregister
- key exchange & MR(Memory Region) management

Datapath

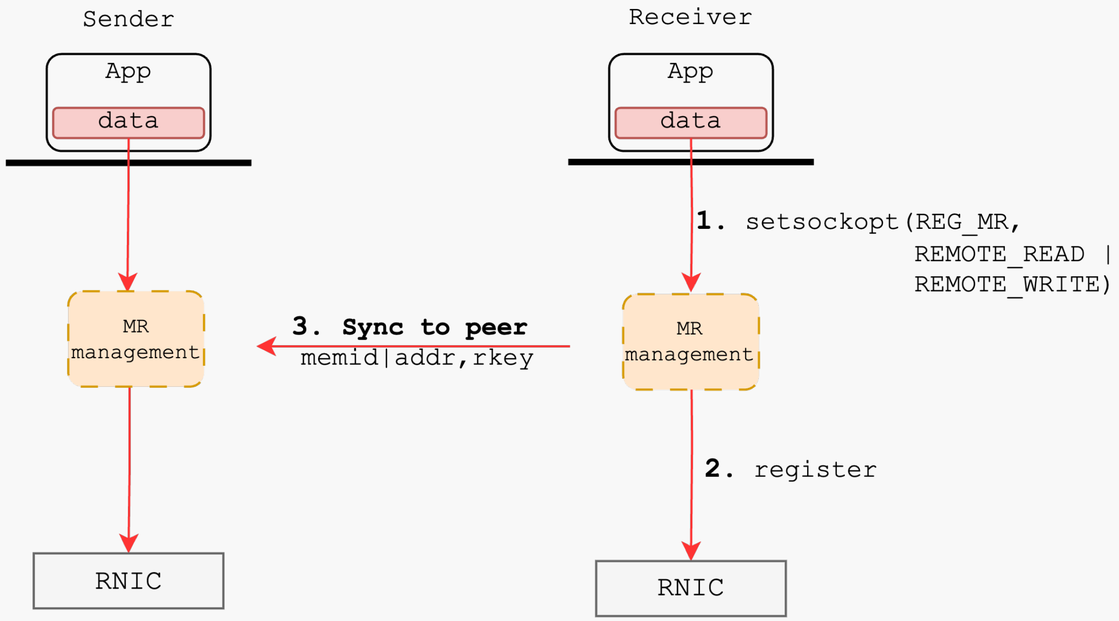
- send
- recv

SMC-R Control path



Handshake to check if both side support ERM

ERM Control path API



REG_MR

```
struct smc_erm_cmd_reg_mr regmr = {
    .type = SMC_ERM_CMD_REG_MR,
    .addr = buff,
    .len  = buff_size,
    .access_flags = SMC_ACCESS_LOCAL_WRITE |
                    SMC_ACCESS_REMOTE_READ |
                    SMC_ACCESS_REMOTE_WRITE,
};
memid = setsockopt(sock, SOL_SMC, SMC_ERM_CMD, &regmr, sizeof(regmr));
```

DEREG_MR

```
struct smc_erm_cmd_dereg_mr deregmr = {
     .type = SMC_ERM_CMD_DEREG_MR,
     .memid = memid,
};
ret = setsockopt(conn->sock, SOL_SMC, SMC_ERM_CMD, &deregmr, sizeof(deregmr));
```

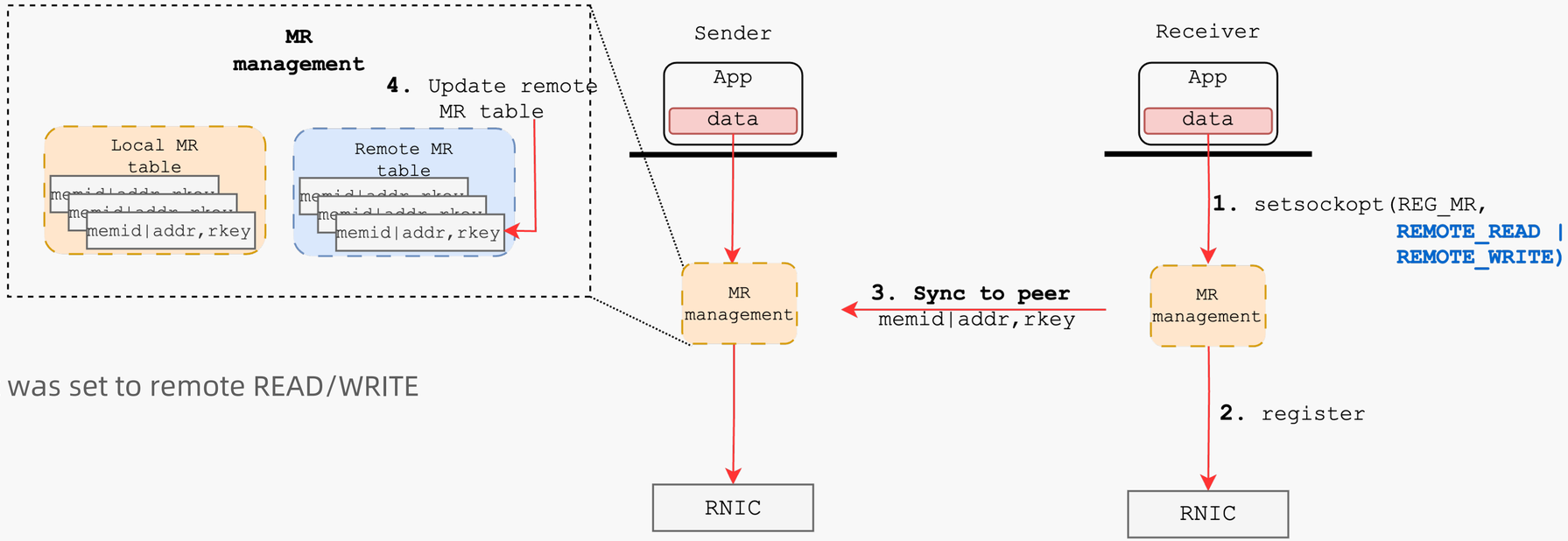
MR management

Memory Registration

- Same as RDMA user memory registration

MR management

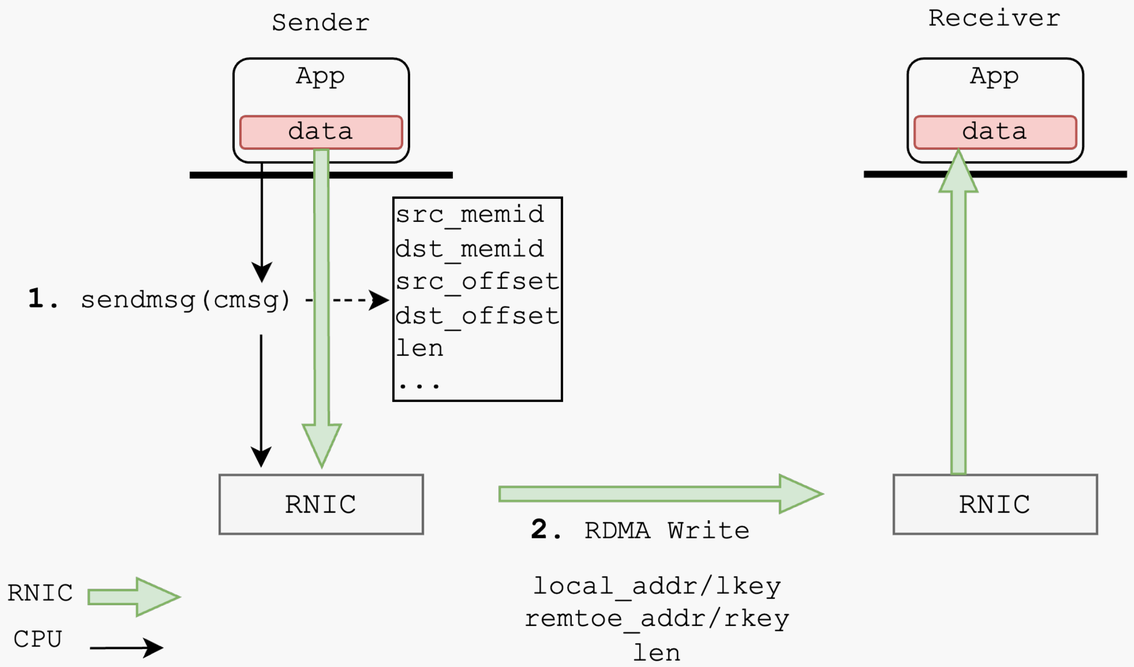
1. MR managed in kernel space
2. MR is synchronized to Peer if it was set to remote READ/WRITE



REG_MR Flow

1. Receiver register MR with REMOTE_READ/WRITE set
2. Register the memory locally
3. ERM will sync the new entry to the Sender
4. Sender update it's "Remote MR table", so the sender can do Remote R/W using this memid

ERM data path API



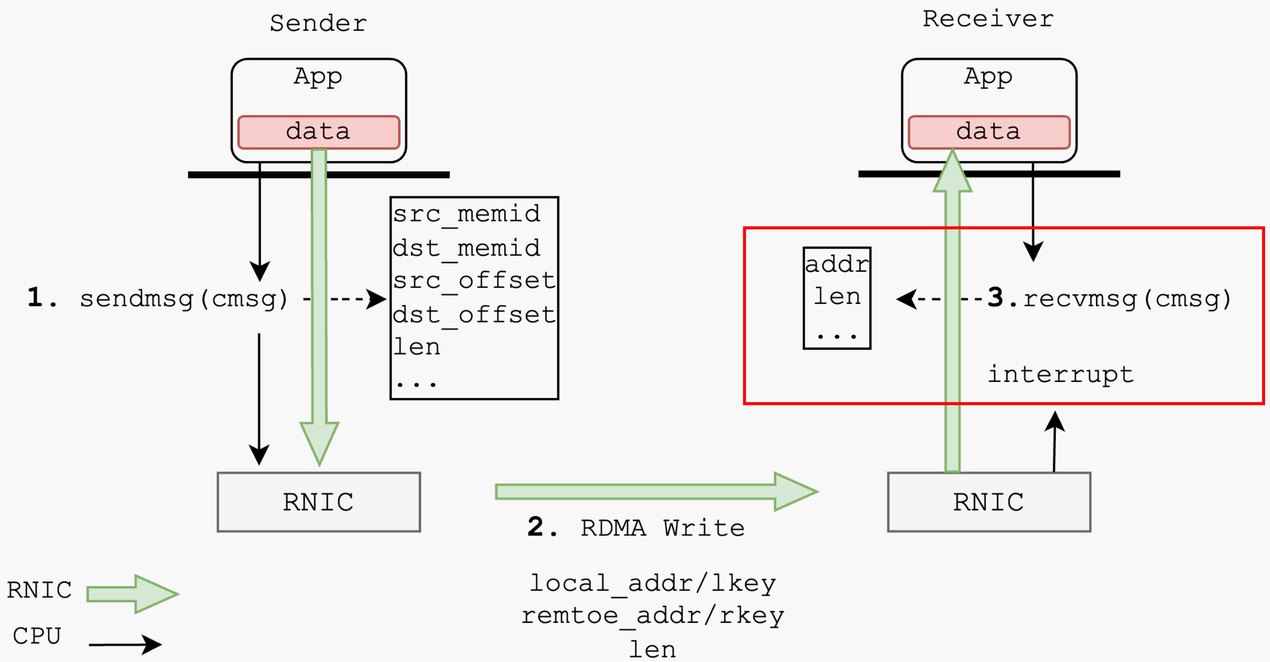
sendmsg

```
struct smc_erm_cmd_copy copycmd = {
        .hdr.type     = SMC_ERM_CMD_COPY,
        .hdr.flags    = SMC_ERM_CMD_FLAG_REMOTE_COMPL,
        .len          = msg_size,
        .src_idx      = local_memid,
        .src_offset   = 0,
        .dst_idx      = remote_memid,
        .dst_offset   = 0,
};
struct msghdr msgh = {
        .msg_control          = u.buf,
        .msgh.msg_controllen = sizeof(u.buf),
};
struct cmsghdr *cmsg = CMSG_FIRSTHDR(&msgh);
cmsg->cmsg_level      = SOL_SMC;
cmsg->cmsg_type       = SMC_ERM_CMD;
cmsg->cmsg_len        = CMSG_LEN(sizeof(struct smc_erm_cmd_copy));
memcpy(CMSG_DATA(cmsg), &copycmd, sizeof(copycmd));

sendmsg(conn->sock, &msgh, 0);
```

Alibaba Cloud

ERM data path API



recvmsg

```
ret = recvmsg(conn->sock, &msgh, MSG_ERRQUEUE);

for (cmsg = CMSG_FIRSTHDR(&msgh); cmsg != NULL; cmsg = CMSG_NXTHDR(&msgh, cmsg)) {
        if (cmsg->cmsg_level != SOL_SMC ||  cmsg->cmsg_type != SMC_ERM_CMD_COMPL)
                continue;

        struct smc_rmem_compl_event *ev = (struct smc_rmem_compl_event *)CMSG_DATA(cmsg);
        switch (ev->event) {
        case SMC_ERM_CMD_COPY:
                data       = ev->copy.addr;
                recv_len = ev->copy.len;
                break;
        default:
                error("un insterested event: %d\n", ev->event);
                break;
        }
}
```

Alibaba Cloud

03. Preliminary results

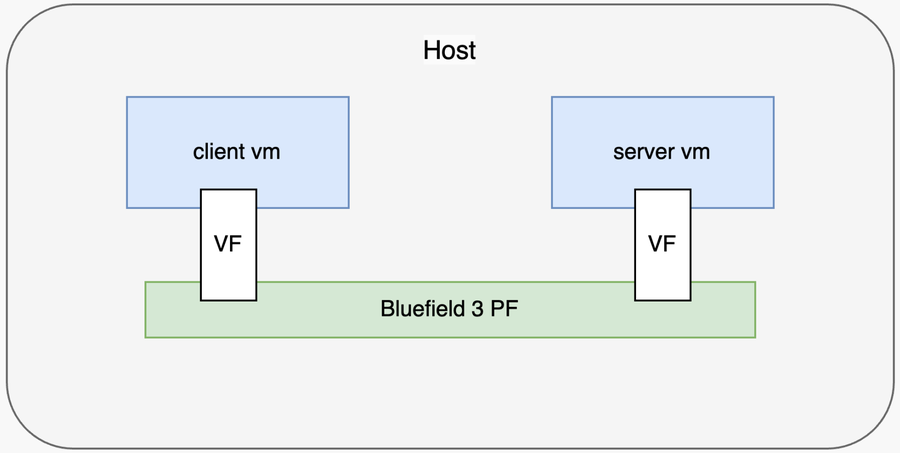
Test setup and Performance

Hardware:

- CPU: Intel(R) Xeon(R) Platinum 8469C CPU @ 2.6GHz, SPR
- Memory: 2T(64GB*32), DDR5, 4800 MT/s
- NIC: Mellanox Bluefield 3, 200Gbps *2, with PCIe 5.0 x16

Topology:

- 2 VM’s running on the same host
- Both VM’s memory are on the same Numa Node
- Both VM’s CPU are in the same socket
- 2 VFs are on the same PF，each allocate to 1 VM

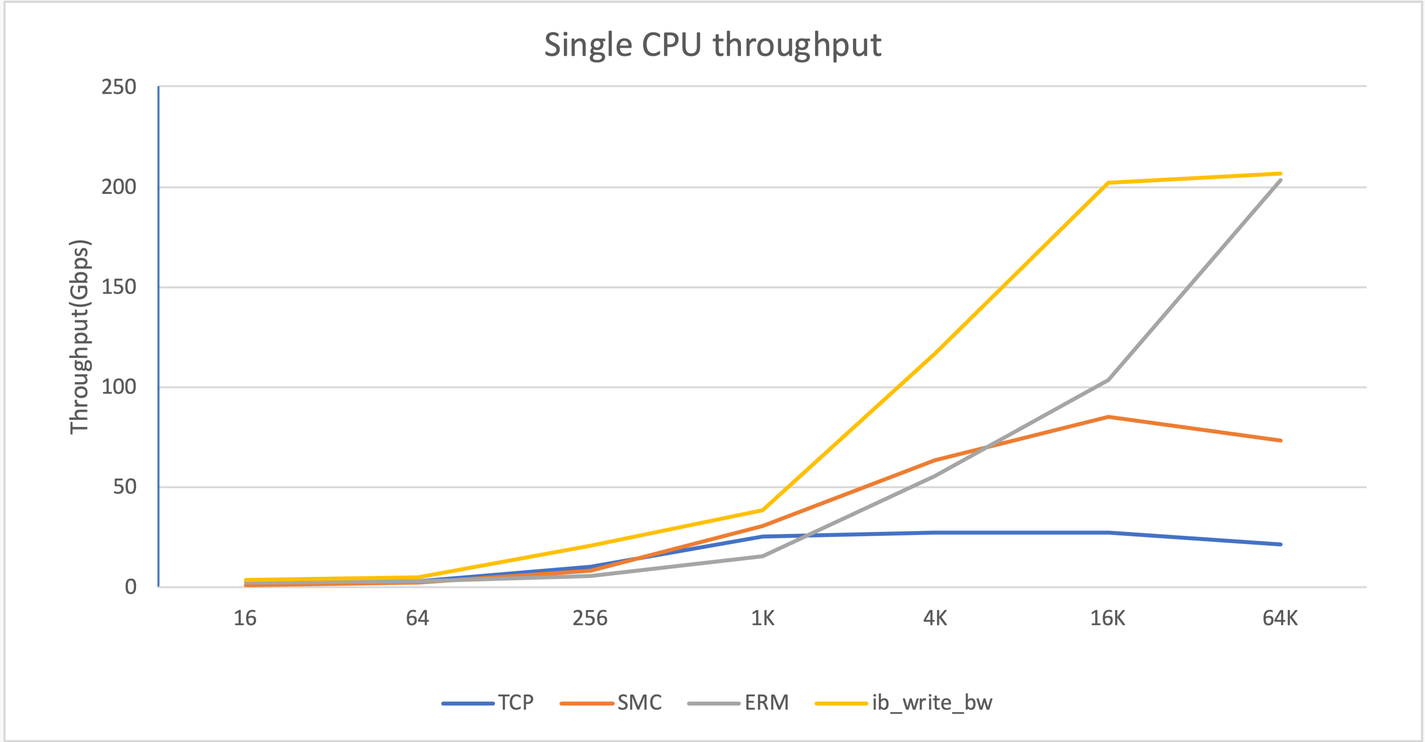


Alibaba Cloud

Performance

Throughput Test

- Data from sender to receiver only
- Both ERM and userspace RDMA (ib_write_bw) can saturate 200Gbps with 1 core



Alibaba Cloud

04. Status + Future work

Alibaba Cloud

Status

- Still a POC, many code are hardcoded
- Handshake protocol support
- Complete the RFC and send to netdev

Alibaba Cloud

Future work

Busypoll support

- Busypoll is important for extreme low latency
- SMC don’t support busypoll now
- Application can busypoll memory in userspace
- TCP busypoll may “do work for others”, same for SMC

Combine io-uring ?

- High performance zerocopy requires asynchronous API
- io-uring is asynchronous by nature

Alibaba Cloud

05. Q & A

Alibaba Cloud

COMPUTING FOR THE VALUE
BEYOND COMPUTATION

Alibaba Cloud

Open questions

Where to put the cmsg ? errqueue or recv queue ?

- errqueue like MSG_ZEROCOPY, add extra complexity to userspace
- recv queue without IOV is a bit wired