jVerbs: RDMA support for the JVM

Patrick Stuedi, Bernard Metzler, Animesh Trivedi
IBM Research Zurich

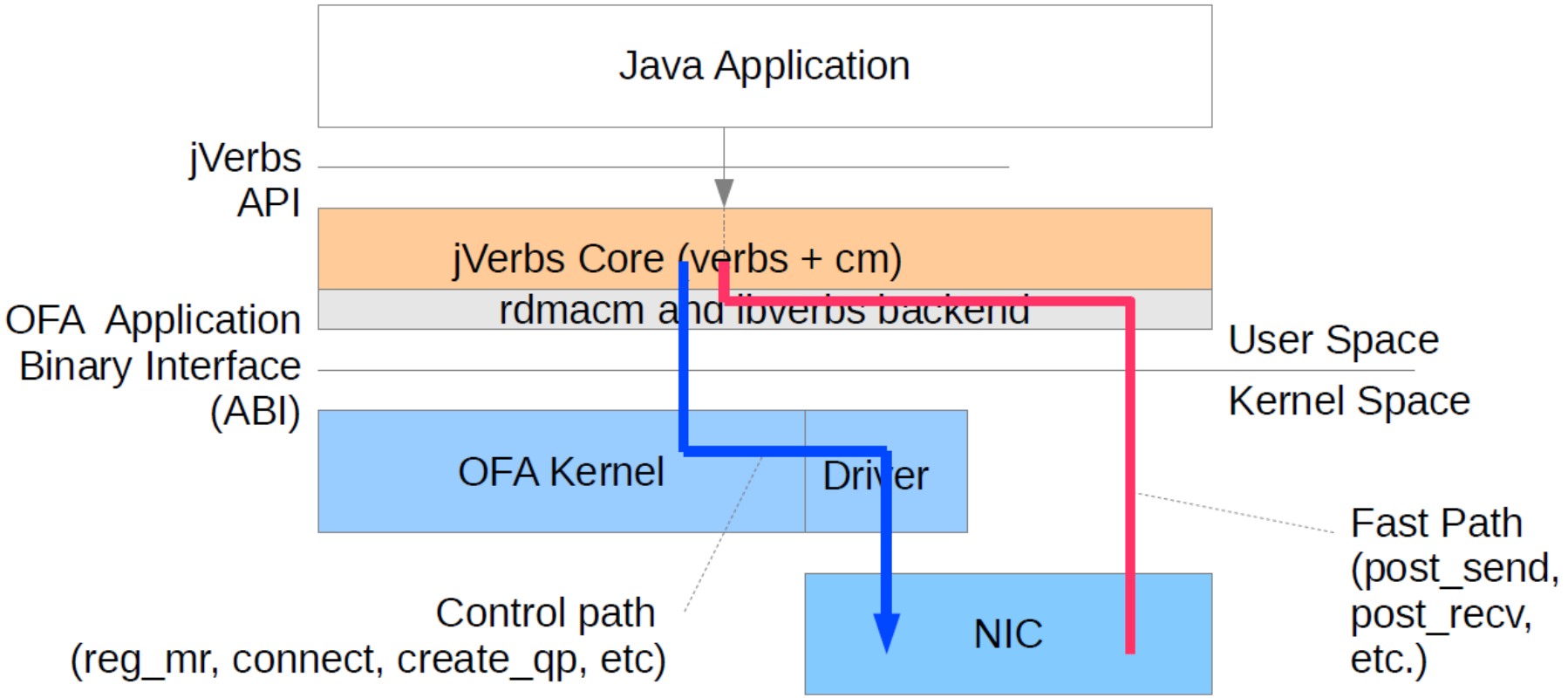
jVerbs: RDMA Support for the Java Virtual Machine

- Native RDMA support for the JVM
  - Set of APIs for programming RDMA in Java
  - Part of IBM Java SE Version 7:
    http://www.ibm.com/developerworks/java/jdk/linux/download.html
  - Documentation:
    http://www-01.ibm.com/support/knowledgecenter/SSYKE2_7.0.0/com.ibm.java.lnx.70.doc/diag/understanding/rdma_jverbs.html
  - Publication: SOCC'13
    http://dl.acm.org/citation.cfm?id=2523631

Provides two sets of APIs

- **Verbs API:**
  - Java counterpart of ibverbs
  - Object oriented
- **Endpoint API:**
  - High level interface
  - Connection management interface like sockets (e.g., endpoint.connect())
  - Data path like verbs (e.g., endpoint.postSend(wr))

jVerbs Architecture



Example: Programming with the jVerbs Verbs API

- Create protection domain and register memory:

```
ProtectionDomain pd = context.allocProtectionDomain();
ByteBuffer dataBuf = ByteBuffer.allocateDirect(1024);
MemoryRegion mr = pd.registerMemoryRegion(dataBuf, access).execute().getMemoryRegion();
```

- Create queue pair:

```
ConnectionId id = ConnectionId.create(cmChannel, PortSpace.RDMA_PS_TCP);
QueuePairInitAttribute attr = new QueuePairInitAttribute();
attr.setQueuePairType(Type.IBV_QPT_RC);
attr.setReceiveCompletionQueue(cq);
attr.setSendCompletionQueue(cq);
QueuePair qp = id.createQueuePair(pd, attr);
```

- Post recv operation:

```
LinkedList<ReceiveWorkRequest> wrList_recv = new LinkedList<ReceiveWorkRequest>();
ScatterGatherElement sgeRecv = new ScatterGatherElement();
sgeRecv.setAddress(mr.getAddress());
sgeRecv.setLength(mr.getLength());
sgeRecv.setLocalKey(mr.getLocalKey());
LinkedList<ScatterGatherElement> sgeListRecv = new LinkedList<ScatterGatherElement>();
sgeListRecv.add(sgeRecv);
ReceiveWorkRequest recvWR = new ReceiveWorkRequest();
recvWR.setSgeList(sgeListRecv);
recvWR.setWorkRequestId(1000);
wrList_recv.add(recvWR);
PostReceiveMethod postRecv = qp.preparePostReceive(wrList_recv);
postRecv.execute();
```

Example: Programming with the jVerbs Endpoint API

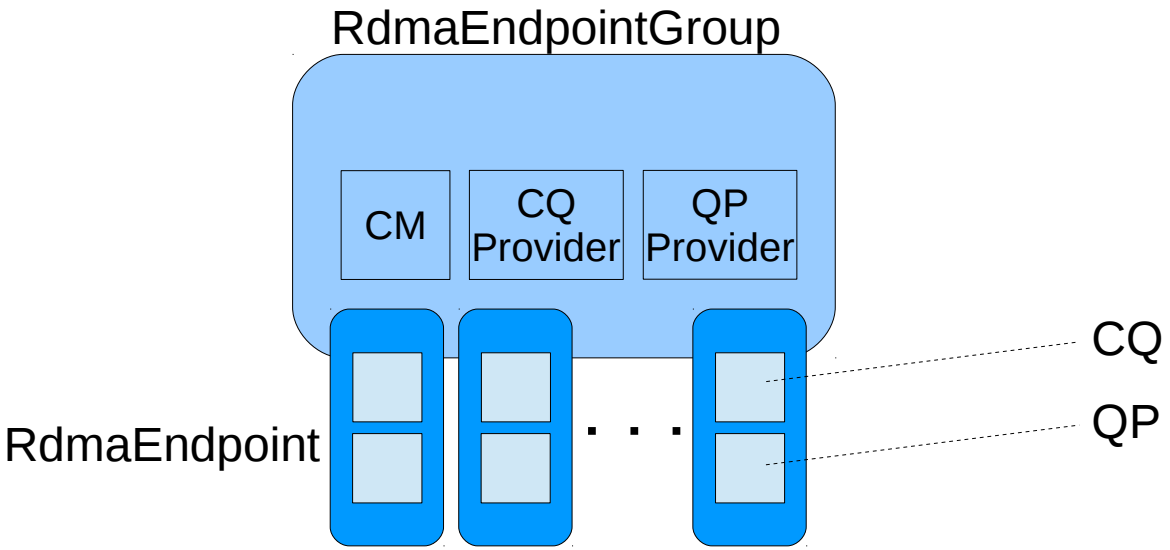
- Create endpoint and connect it:

```
InetSocketAddress dst = new InetSocketAddress(InetAddress.getByName("127.0.0.1"), 1919);
RdmaActiveEndpointGroup group = new RdmaActiveEndpointGroup(this);
RdmaEndpoint endpoint = group.createEndpoint();
ConnectDispatcher connectDispatcher = new ConnectDispatcher();
endpoint.connect(dst, 1000, connectDispatcher);
connectDispatcher.wait();
```

- Post send operation:

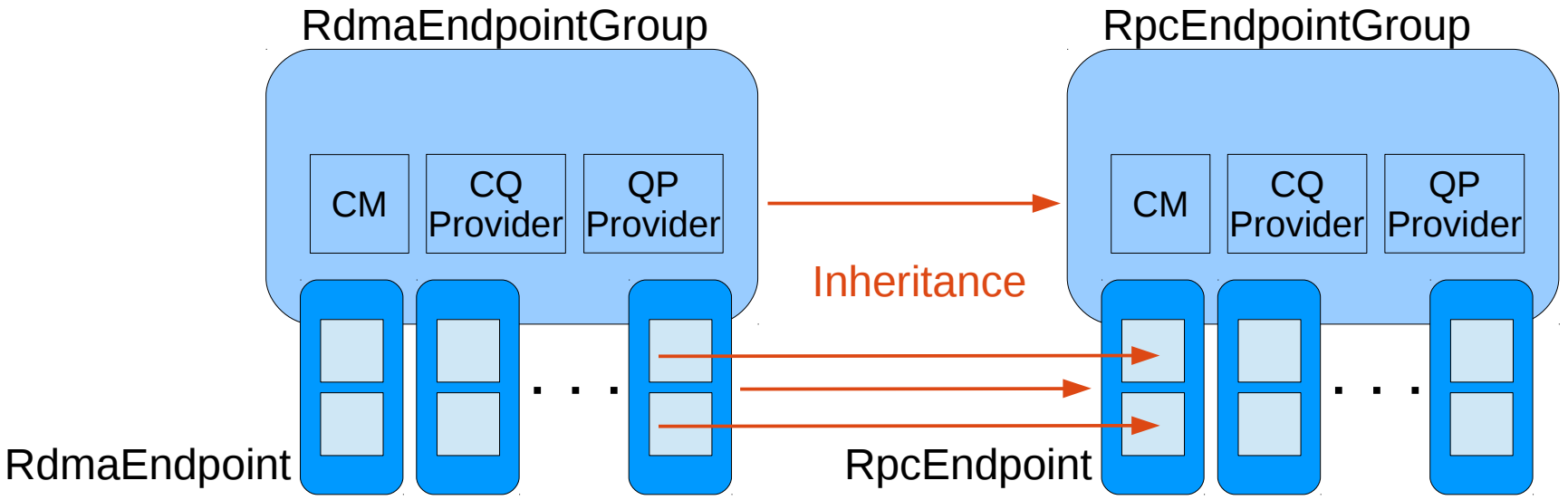
```
LinkedList<SendWorkRequest> wrList_send = new LinkedList<SendWorkRequest>();
ScatterGatherElement sgeSend = new ScatterGatherElement();
sgeSend.setAddress(mr.getAddress());
sgeSend.setLength(mr.getLength());
sgeSend.setLocalKey(mr.getLocalKey());
LinkedList<ScatterGatherElement> sgeList = new LinkedList<ScatterGatherElement>();
sgeList.add(sgeSend);
SendWorkRequest sendWR = new SendWorkRequest();
sendWR.setWorkRequestId(1001);
sendWR.setSgeList(sgeList);
sendWR.setOpcode(Opcode.IBV_WR_RDMA_READ);
sendWR.getRdma().setRemoteAddress(remoteMr.getAddress());
sendWR.getRdma().setRemoteKey(remoteMr.getLocalKey());
wrList_send.add(sendWR);
PostSendMethod postSend = endpoint.preparePostSend(wrList_send);
postSend.execute();
```

RDMA Groups and Endpoints



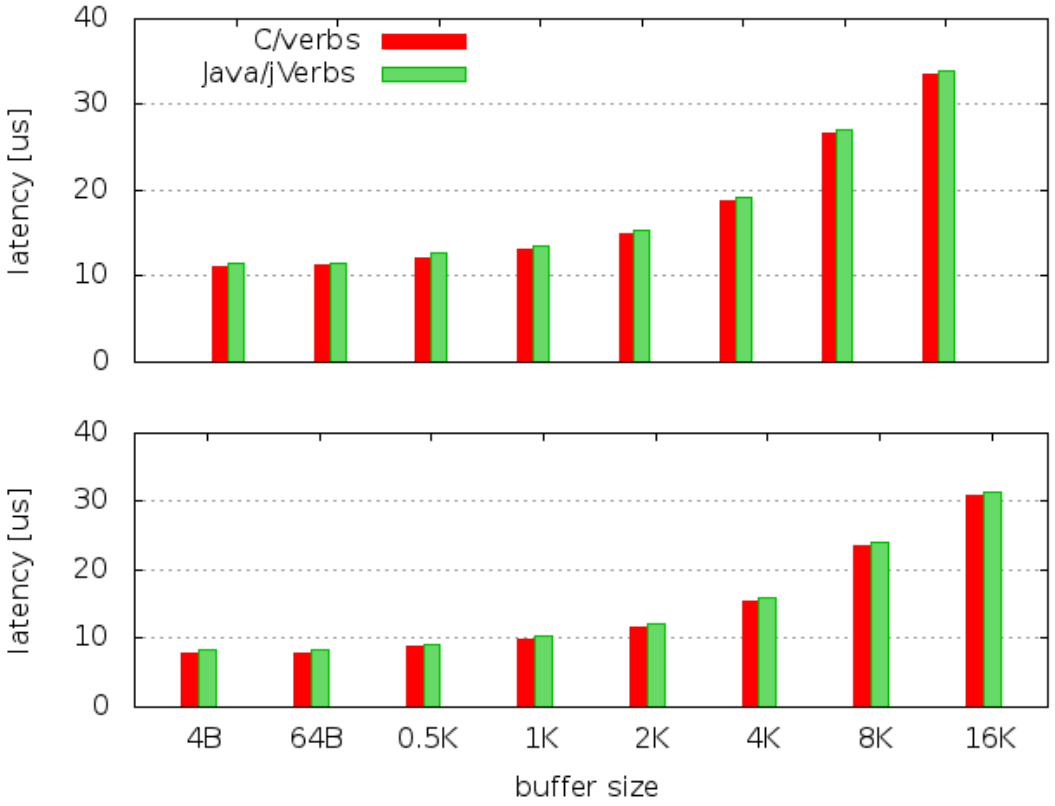
- RdmaEndpointGroup:
  - Creates endpoints (factory for endpoints)
  - Provides resources (CQ, QP) for endpoints (factory for resources)
  - Manages state of endpoints (CM event management)
- Application passes endpoint properties when creating RdmaEndpointGroup
  - CQ property: polling versus blocking, shared CQ versus individual CQ
  - QP property: size
- All endpoints belonging to the same group share the same properties

Group Inheritance: Custom RDMA Groups



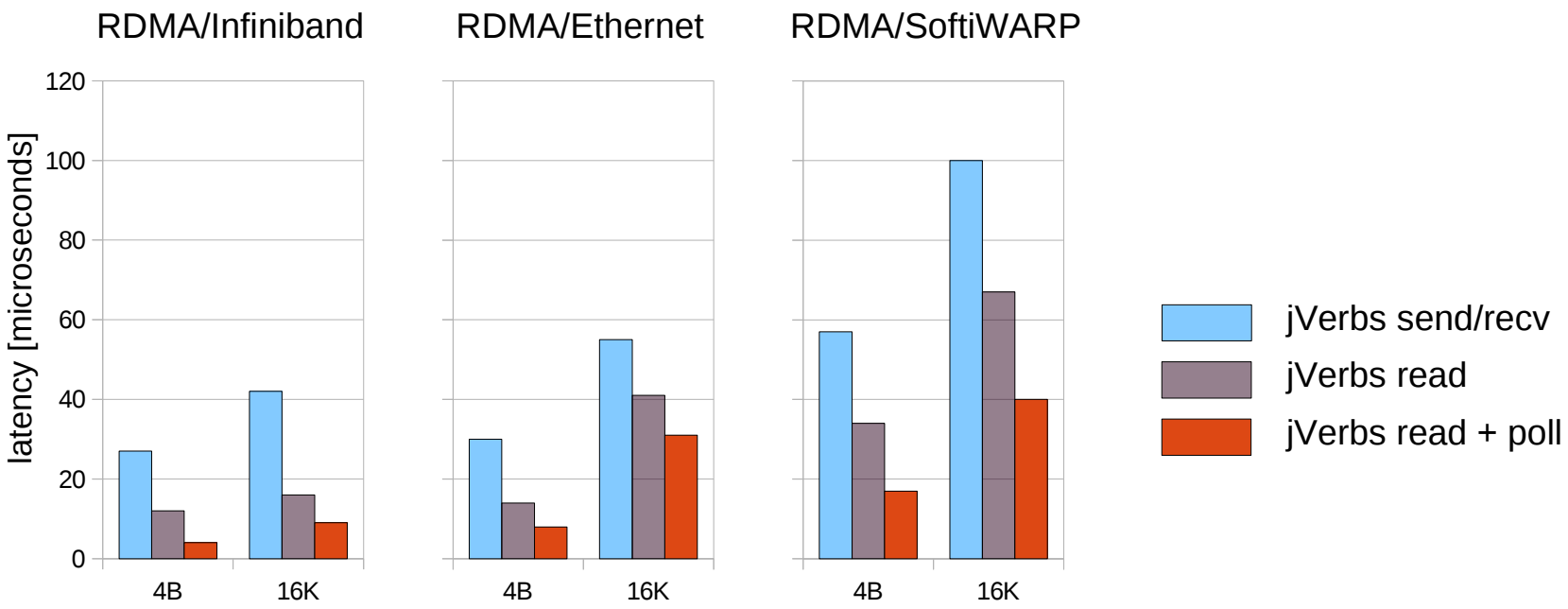
- Groups can be extended using inheritance
- Extended groups allow creating custom endpoints tailored to specific applications
- Example: RPCEndpointGroup
  - Provides NUMA aware event processing
  - Provides array of CQs allocated to match the NUMA layout
  - RpcEndpoints include extra operations for RPC processing

Performance: jVerbs vs ibverbs



- Setup: Two Intel Xeon E5-2690 boxes connected by a Chelsio T4
- (top) send/recv with polling
- (bottom) RDMA/read with polling
- jVerbs performance matches native C performance

Performance: jVerbs on Infiniband and SoftiWARP



- Infiniband Setup: Mellanox ConnectX-2 (QDR)
  - jVerbs Read latency: ~3.5 usec
- IWARP: Chelsio T4 (10 GbE)

Conclusions

- jVerbs exposes RDMA verbs interface of modern high-performance NICs to the JVM
  - Richer semantics than traditional sockets
- Enables ultra-low latency networking in JVM-based distributed applications running in the cloud
- Opens door for more efficient network I/O in distributed systems
- **Thanks!**