# 基于 DPU 芯片的 RDMA atomic 操作的硬件实现

吴小林[1] 王万财[1] 高 志[1]
WU Xiaolin WANG Wancai GAO Zhi

**摘 要** RDMA 网络具有高带宽、低延时、低 CPU 负载的特点，广泛应用于数据中心。RDMA 技术中的 RoCEv2 由于兼容当前数据中心的网络层与数据链路层，而被认为是一种低成本的 RDMA 技术解决方案。介绍了一种 DPU 芯片中 RDMA atomic 操作的硬件实现。DPU 芯片基于 RoCEv2 协议，在网络拥塞出现丢包时，能够实现 RDMA atomic 操作在响应端最多执行一次的要求。在硬件电路功能仿真中，已实现了该需求，且重传 RDMA atomic 操作的延迟相比之前缩短了至少 2 μs。

**关键词** RDMA；RoCEv2；RDMA atomic

doi：10.3969/j.issn.1672-9528.2024.09.039

## 0 引言

当今是云计算、大数据的时代，企业业务持续增长，需要存储系统的 IO 性能也持续增长。传统的 TCP/IP 技术在数据包的处理过程中，要经过操作系统及其他软件层，数据在系统内存、处理器缓存和网络控制器缓存之间来回进行复制，给服务器的 CPU 和内存造成了沉重负担。尤其是网络带宽、处理器速度与内存带宽三者的严重“不匹配性”，更加剧了网络延迟效应。

为了降低数据中心内部网络延迟、提高处理效率，远程直接内存访问（remote direct memory access，RDMA）技术应运而生。RDMA 是一种高速网络传输技术，传输数据时可以绕过操作系统内核直接对远端内存进行读写。相较于传统的 TCP/IP 网络，RDMA 具有低延迟、高吞吐量、低 CPU 负载的特点，更适合数据中心的网络传输需求 [1-2]。

1. 成都北中网芯科技有限公司 四川成都 611731

未来，随着智能电网技术的不断进步和数据挖掘技术的深入发展，基于数据挖掘的供电所错误接线分析方法将会得到更加广泛的应用和推广。这不仅可以提高供电所的运维效率和质量，还可以为电力系统的稳定运行和用户的用电安全提供更加坚实的保障。

**参考文献：**

[1] 赵俊红 . 对计量电流互感器常见错误接线的分析与判断 [J]. 电气技术与经济 ,2024(3):199-200+203.
[2] 邵浚哲 , 王乐 , 檀林青 , 等 . 电压回路接线错误导致工频变化量阻抗保护误动作分析 [J]. 电工技术 ,2024(3):124-126+129.
[3] 陈胜 , 刘艳 , 贾宏伟 . 并联式高压电能计量仿真接线培训柜的研究 [J]. 农村电气化 ,2022(10):79-82.
[4] 何程 , 杨红平 , 朱贺 , 等 . 一起 35 kV 专线用户电能计量装置错误接线的分析与研究 [J]. 电工技术 ,2024(2):146-148.
[5] 张贝贝 , 郭左 , 张恒伟 , 等 . 一起错误接线造成保护误动作跳闸事故分析 [J]. 农村电工 ,2024,32(1):58-59.
[6] 周凯欣 , 冯萧飞 , 苏盛 , 等 . 基于时序关联特性的错误接线漏电用户定位方法 [J]. 仪器仪表学报 ,2023,44(10):247-259.
[7] 左素梅 , 左磊 , 杨欣 . 一起错误接线导致线损异常事例分析 [J]. 农村电工 ,2023,31(5):51.

**【作者简介】**

杨宸（1995—），男，甘肃白银人，本科，助理工程师，研究方向：电力营销。

张文杰（1999—），男，甘肃平凉人，本科，助理工程师，研究方向：电气工程。

任启涛（1997—），男，甘肃平凉人，本科，助理工程师，研究方向：电力营销。

魏凯（1990—），男，甘肃金昌人，本科，工程师，研究方向：电力系统终端互动调节。

朱博（1999—），男，甘肃平凉人，本科，助理工程师，研究方向：电气工程及其自动化。

（收稿日期：2024-06-20）

RDMA 技术的实现需要遵循 InfiniBand（IB）协议 [3]。该协议作为一个新一代网络协议，需要依靠专门的硬件环境来支撑。IB 协议在数据链路层与网络层对于传统以太网协议和 IP 协议的不兼容，限制了 RDMA 技术只能应用在私有云和一些定制服务的数据中心 [4]，因此在后续的 IB 协议版本中提出了 RDMA 融合以太网（RDMA over converged ethernet，RoCE）协议。

RoCE 协议首次提出将 IB 协议的传输层承载到以太网的数据链路层。RoCEv2 针对 RoCE 协议进行了一些改进，使用了 UDP+IP 作为网络层，使得数据包可以在以太网链路上被路由。然而，相比面向连接的 TCP 协议，基于无连接的 UDP 协议不像 TCP 协议那样有滑动窗口、确认应答等机制来实现可靠传输。一旦网络拥塞出现丢包，RoCEv2 发送端只能采用 Go-Back-N 的重传机制来进行重传。RoCEv2 响应端为了提高 RDMA 命令的处理效率、降低延迟，提出 RDMA 命令在响应端只执行一次的要求。本文介绍的就是 RoCEv2 响应端对 RDMA atomic 命令的具体硬件实现。

## 1 RDMA 的工作队列

RDMA 技术中，QP（queue pair）作为 RDMA 通信的基本单元，可以在两个节点之间进行直接内存访问，实现零拷贝的数据传输和低延迟的交互。如图 1 所示，QP 通常包含三种工作队列：发送队列（send queue，SQ）、接收队列（receive queue，RQ）、完成队列（complete queue，CQ）。SQ 用于存储发送方的数据和命令信息，RQ 用于存储接收方的数据和命令信息，CQ 用于存储数据传输完成的相关信息，可以多个 SQ 或多个 RQ 共用一个 CQ。

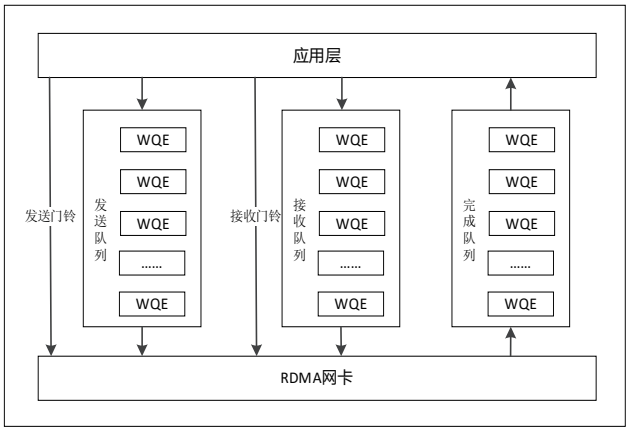


图 1 RDMA 的工作队列

工作队列分别由内存中的多个非连续的区域组成，每个区域按顺序存储着应用层给硬件下发的工作任务（work queue entry，WQE），WQE 中包含两端数据交互的一些信息，具体如图 2 所示。应用层通过门铃机制通知 RDMA 网卡有新的 WQE 存放在工作队列中。

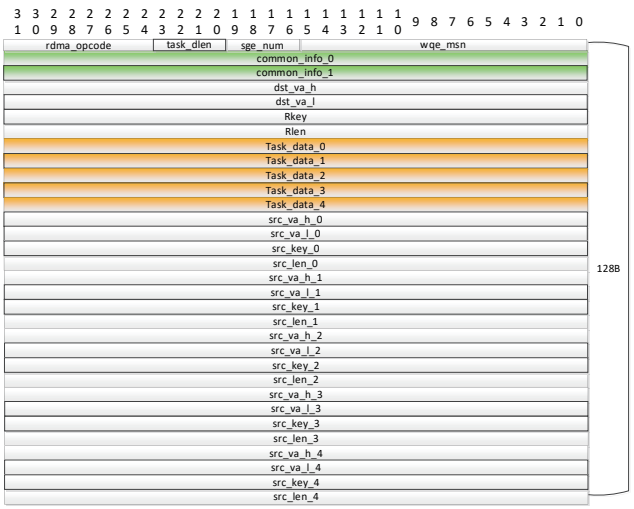


图 2 WQE 的数据格式

## 2 RDMA 的基本操作

RDMA 技术的通信模型如图 3 所示 [5]。纵向上划分为控制通路、数据通路。控制通路需要进入内核态，准备通信所需的各种资源，需要 CPU 的参与，一般在链路建立时使用。控制通路通过发送 RDMA Send-Recv 命令，创建和配置 RDMA 通信的基本元素，如 QP、CQ 等。数据通路专门负责数据的收发，因为已通过控制通路获取到了通信所需要的基本元素，如虚拟地址 VA、远端的 Rkey 等，此时进行数据通信不需要 CPU 介入，可以直接对远端内存进行操作。

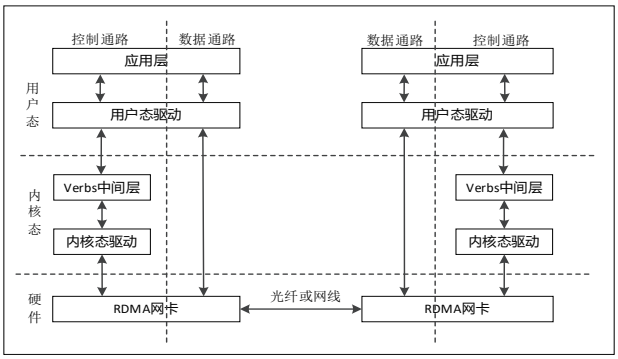


图 3 RDMA 技术的通信模型

RDMA 通信模型中，控制通路进行 RDMA Send-Recv 操作的流程如图 4 所示，描述如下。

（1）远端应用层往 RQ 中下发 WQE，并通过接收门铃通知硬件有新的 WQE 产生。

（2）本地应用层往 SQ 中下发 WQE，该 WQE 中的任务为 RDMA Send-Recv 类型。

（3）本地 RDMA 网卡收到软件下发的发送门铃，从 SQ 中获取 WQE，进行解析。

（4）本地 RDMA 网卡根据 WQE 信息，从本地内存中读取数据，组装成数据包发送。

（5）本地 RDMA 网卡将数据包，通过物理链路发送给

远端的 RDMA 网卡。

（6）远端 RDMA 网卡接收并解析数据包，并读取 RQ 中的 WQE，将解析后的数据放到 WQE 指定的位置。

（7）远端 RDMA 网卡生成 CQE，放在 RQ 所对应的 CQ 中。

（8）远端 RDMA 网卡，将接收数据的处理结果，回复 ACK/NAK 报文给发送端。

（9）本地 RDMA 网卡收到 ACK/NAK 报文后，生成 CQE，放到 CQ 中。

（10）本地应用层收到命令处理完成的指示。

（11）远端应用层收到接收到新数据的指示。

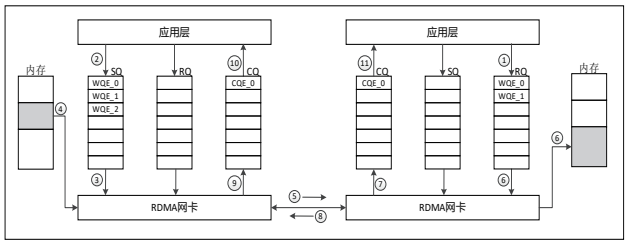


图 4 RDMA Send-Recv 操作流程

RDMA 通信模型中，数据通路进行 RDMA Write 操作的流程如图 5 所示，描述如下。

（1）本地应用层往 SQ 中下发 WQE，该 WQE 中的任务为 RDMA Write 类型。

（2）本地 RDMA 网卡收到软件下发的发送门铃，从 SQ 中获取 WQE。

（3）本地 RDMA 网卡解析 WQE，将 WQE 中包含的 src_va，进行地址转换成物理地址，并从该物理地址对应的内存中读取数据，组装成数据包发送。

（4）本地 RDMA 网卡将数据包，通过物理链路发送给远端的 RDMA 网卡。

（5）远端 RDMA 网卡接收数据包，将数据包中的 dst_va，进行地址转换成物理地址，并将数据放到该物理地址对应的内存。

（6）远端 RDMA 网卡，将接收数据的处理结果，回复 ACK/NAK 报文给发送端。

（7）本地 RDMA 网卡收到 ACK/NAK 报文后，生成 CQE，放到 CQ 中。

（8）本地应用层收到命令处理完成的指示。

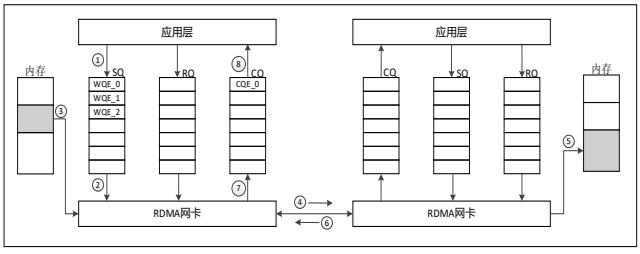


图 5 RDMA Write 操作流程

RDMA 通信模型中，数据通路进行 RDMA Read 操作的流程如图 6 所示，描述如下。

（1）本地应用层往 SQ 中下发 WQE，该 WQE 中的任务为 RDMA Read 类型。

（2）本地 RDMA 网卡收到软件下发的发送门铃，从 SQ 中获取 WQE。

（3）本地 RDMA 网卡解析 WQE，将 WQE 中包含的 dst_va，封装到读请求数据包，并通过物理链路发送给远端的 RDMA 网卡。

（4）远端 RDMA 网卡接收数据包，将数据包中的 dst_va，进行地址转换成物理地址，并从该物理地址对应的内存中读取数据，封装成读响应数据包。

（5）远端 RDMA 网卡，将读响应数据包，通过物理链路发送给本地 RDMA 网卡。

（6）本地 RDMA 网卡收到读响应数据包后，再次读取本次操作对应的 WQE，将 WQE 中包含的 src_va，进行地址转换成物理地址，并将读响应数据包中的数据放到该物理地址对应的内存。

（7）本地 RDMA 网卡，生成 CQE，放到 CQ 中。

（8）本地应用层收到命令处理完成的指示。

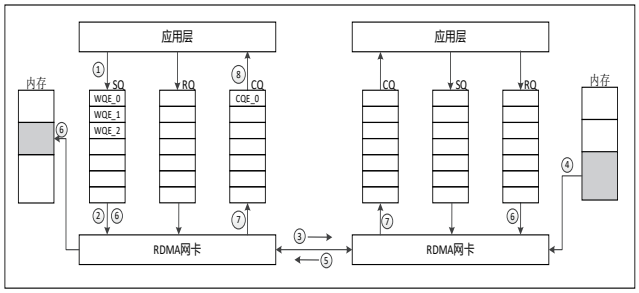


图 6 RDMA Read 操作流程

## 3 RDMA 的原子操作

在 RDMA 数据通路，以 RDMA Write 和 RDMA Read 为基础，RDMA 网卡还支持 RDMA atomic 操作。RDMA atomic 是对远端内存中的 64 bit 数据，执行 Read+Modify+Write 的组合操作，且读写操作之间不能被打断。在 IB 协议中，当前支持两种原子操作：Compare and Swap（简称 CAS）、Fetch and Add（简称 FAA）。两种原子操作的说明如表 1 所示。IB 协议规定，一次 RDMA atomic 操作，在响应端最多执行一次，并将执行结果保存在响应端。后续即使网络丢包或响应端异常导致发送端重传，响应端不需要再执行 RDMA atomic 操作，直接返回之前保存的执行结果即可。该规定不仅可以提高响应端处理 RDMA 命令的效率，从整体上降低 RDMA 的延迟，对于 Fetch and Add 的 atomic 操作，还能保证应用层命令正确执行。

表 1 RDMA atomic 操作介绍

| 原子操作类型 | WQE 中的操作数 | 操作说明 |
|---|---|---|
| Compare and Swap | 操作数 1：虚拟地址 VA；<br>操作数 2：被比较数；<br>操作数 3：内存更新值。 | 1. 读取远端 VA 地址的 64 bit 数据；<br>2. 将读取的 64 bit 数据，返回请求端；<br>3. 将读取的 64 bit 数据，与操作数 2 比较；<br>4. 若 2 中的比较结果一致，将操作数 3 写入远端 VA 地址对应的内存；否则不写入。 |
| Fetch and Add | 操作数 1：虚拟地址 VA；<br>操作数 2：递增数。 | 1. 读取远端 VA 地址的 64 bit 数据；<br>2. 将读取的 64 bit 数据，返回请求端；<br>3. 将读取的 64 bit 数据，加上操作数 2，重新写入远端 VA 地址对应的内存。 |

以 Fetch and Add 的 atomic 操作为例，一次正常的 RDMA atomic 操作的交互，如图 7 所示。

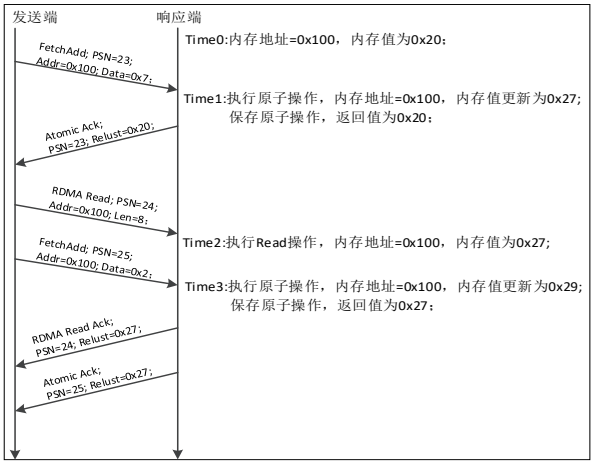


图 7 正常的 RDMA atomic（FAA）操作

以 Fetch and Add 的 atomic 操作为例，一次带有重传的 RDMA atomic 操作的交互，如图 8 所示。

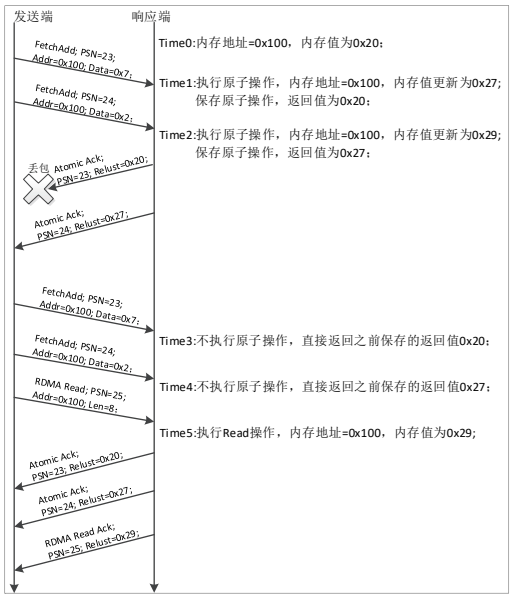


图 8 重传的 RDMA atomic（FAA）操作

## 4 RDMA 原子操作的硬件实现

为确保一次 RDMA atomic 操作在响应端最多执行一次，响应端需存储已正确执行的 atomic 命令的结果。在 IB 协议中，发送端并发的 atomic 命令数，是在 RDMA 建链期间与响应端协商的。若要将 atomic 命令执行结果缓存在硬件中，则需要按最大并发数进行设计，这会导致硬件最终实现面积过大。

本项目参考了 RDMA 技术中 SQ/RQ/CQ 的实现原理，在主机内存中虚拟出来一个工作队列 AQ（ACK queue）来存储每个 atomic 命令的执行结果、AQ 对应的 AQC（ACK queue context）存储 AQ 相关的参数。假设 RDMA 建链期间协商的 atomic 命令数是 256，则 AQ 需存储 256 个 atomic 命令的执行结果（AQ entry，AQE），且每 64 个 AQE 一组，将 PSN 的起始位置在 AQC 中进行存储。如此硬件在收到一个 atomic 命令后，通过 AQC 中 AQ 的分组信息，就可以缩小查找范围，避免出现多次、大数据量地从主机内存读取数据（最恶劣情况下，需要将 AQ 中所有的 AQE 读出），造成 atomic 操作的延迟增加。主机内存中 AQ 及 AQC 的数据结构说明如图 9 所示。

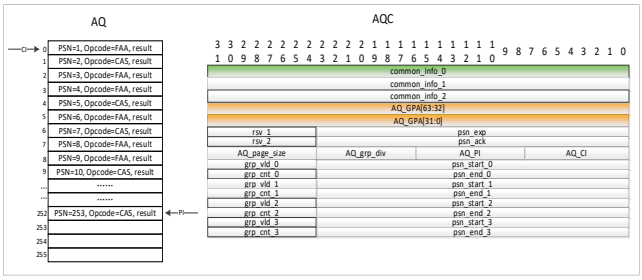


图 9 AQ 及 AQC 数据结构

因 AQ 在主机内存中是连续存放的，根据 AQC 中 AQ 的起始 GPA、AQC 中每组包含的 AQE 个数 AQ_grp_div，就可以计算出来每个 grp 的起始 GPA。同时，因发送端 RDMA 命令的 PSN 连续，响应端根据当前 RDMA 命令的 PSN，即可预期下一个 RDMA 命令的 PSN。基于以上两点，硬件收到一个 atomic 命令的处理流程如图 10 所示。

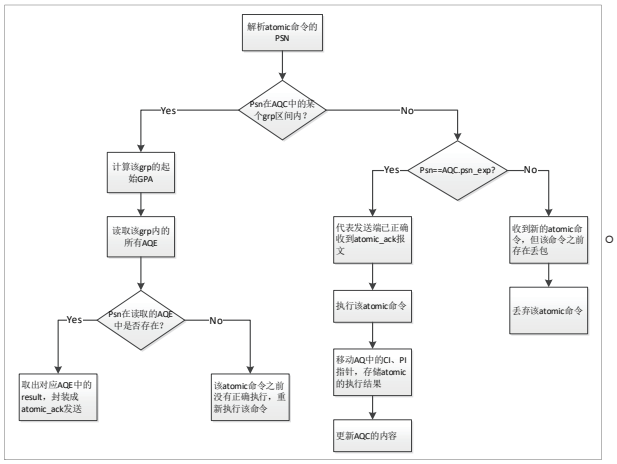


图 10 RDMA atomic 命令的处理流程

需要注意的是，响应端在执行 atomic 命令的过程中，若该 atomic 命令的合法性检查未通过，则该 atomic 命令被丢弃，不能正确执行。该 atomic 命令后面的其他 RDMA 命令，也需要一并丢弃。发送端需重传出错的 atomic 命令及其后的其他命令（Go-Back-N 重传机制），响应端只有重新正确执行该 atomic 命令之后，才能处理其后的其他命令。该处理的目的旨在确保，对于发送端而言，收到的 ACK 报文是顺序的。

## 5 RDMA 原子操作的功能仿真

本项目中原子操作的功能仿真结果，如图 11 所示。

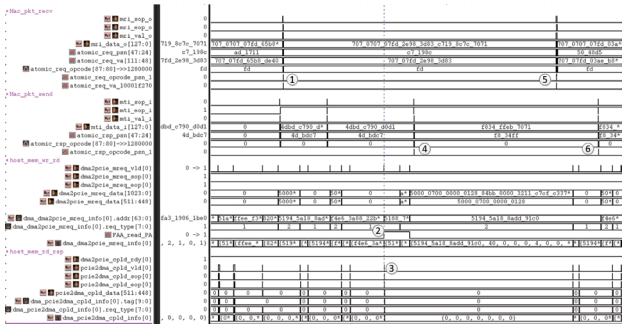

图 11 RDMA atomic 操作的功能仿真结果

具体说明如下。

（1）时刻点 1，从 eth 口接收到一个 RDMA atomic 命令，该命令 opcode=0x14（FAA 类型），va=0x1000_1f270，swapadd_data=0x363_28fe_883a_1230。

（2）时刻点 2，硬件查地址转换表，将 atomic 报文中的 va 转换成 pa，转换出来的 pa=0x5188_7fa3_1906_1be0，之后硬件产生一个 pcie_mem_rd 操作，从该 pa 地址读取 8 B 数据。

（3）时刻点 3，主机读取 pa 地址的内存数据 mem_data=0x7776_7574_7372_7170，通过 pcie_mem_wr 返回给硬件。硬件收到数据，进行数据运算 mem_data+swapadd_data，得到新数据 0x7AD9_9E72_FBAC_83A0，立即进行 pcie_mem_wr 操作，将新数据写入主机内存的 pa 地址，如图 12 所示。

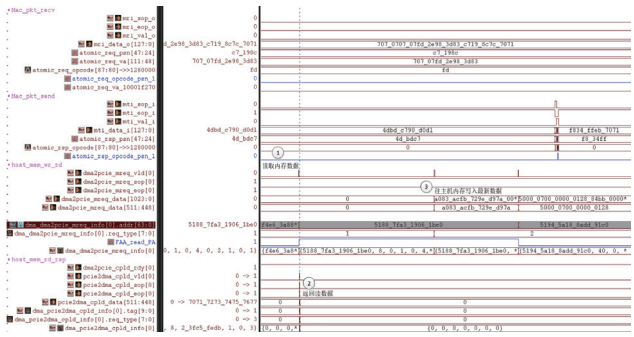

图 12 RDMA atomic 操作的执行过程

（4）时刻点 4，硬件将主机返回的内存数据 mem_data 写入 AQ 中，同时将 mem_data 封装成 atomic_ack 报文，通过 eth 口发送出去。

（5）时刻点 5，硬件返回的 atomic_ack 报文在链路上被丢弃，eth 口收到重传 RDMA atomic 命令。

（6）时刻点 6，硬件从 AQ 中取出上一次 RDMA atomic 命令的执行结果，封装成 atomic_ack 报文，通过 eth 口发送出去。

在硬件工作主频为 900 MHz 时，第一次 RDMA atomic 操作，从收到命令（时刻点 1）到硬件返回 RDMA atomic_ack 报文（时刻点 4）之间的延迟约为 3.19 μs。而重传的 RDMA atomic 操作，从收到重传命令（时刻 5）到硬件返回 RDMA atomic_ack 报文（时刻 6）之间的延迟约为 1 μs。本文中设计的 AQ 机制，在满足协议要求的前提下，有效缩短了 RDMA 命令的处理延迟。

## 6 结语

本文提出了一种 RDMA 技术中 atomic 命令的实现方式。通过设计虚拟工作队列 AQ 来存储 atomic 命令的执行结果，实现了 IB 协议中 atomic 命令在响应端最多执行一次的要求。同时，通过将 AQ 中 AQE 进行分组，通过 2 级查找 + 流水线处理的方式，降低了 atomic 命令的处理延迟。同时，该实现思路，在响应端收到重传的 RDMA read、RDMA write 命令时同样适用（AQE 的数据格式需要调整），具体实现方式在本文中不再赘述。

## 参考文献：

[1]MITTAL R, LAM V T, DUKKIPATI N, et al.TIMELY: RTT-based congestion control for the datacenter[J].Computer communication review, 2015,45:537-550.

[2]ZHU Y, ERAN H, FIRESTONE D, et al.Congestion control for large-scale RDMA deployments[J].Computer communication review, 2015,45:523-536.

[3] InfiniBand™ Architecture Specification Volume 1 Release 1.4[S].[S.l.]:InfiniBand℠ Trade Association, 2020.

[4] 殷建飞 . 基于 RoCE V2 的 SmartNIC 芯片设计与验证 [D]. 西安 : 西安电子科技大学 ,2022.

[5] 刘伟 . Linux 高性能网络详解：从 DPDK、RDMA 到 XDP[M]. 北京 : 人民邮电出版社 ,2023.

## 【作者简介】

吴小林（1986—），女，甘肃兰州人，硕士，中级工程师，研究方向：数字 IC、网络通信、DPU。

王万财（1976—），男，重庆人，硕士，中级工程师，研究方向：数字 IC、网络通信和 DPU、GPU、IC 验证平台。

高志（1984—），男，湖南常德人，硕士，初级工程师，研究方向：数字 IC、智能网卡、DPU。

（收稿日期：2024-06-19）