RDMA 高速网络状态感知与度量指标体系研究

徐佳玮 严 明 吴 杰
(复旦大学计算机科学技术学院 上海 200433)
(复旦大学教育部网络信息安全审计与监控工程研究中心 上海 200433)

摘 要 随着数据中心网络承载的数据量的爆炸式增长,越来越多的框架、模型和应用选择使用 RDMA 技术来加速网络传输。RoCE 是 RDMA 技术在以太网上的实现,但针对 RoCE 网络目前还没有有效的状态感知和度量体系,无法全面展现 RDMA 网络状态。对此,提出针对 RoCE 的实时状态感知系统和多维度多层次的 RDMA 度量指标体系。采用旁路分布式流量捕获方式,运用 Sketch 算法全方位度量 RDMA 网络状态。系统易于部署且成本低,具有可扩展性和灵活性。实验结果表明,该系统能在较低误差下客观反映出 RoCE 网络状态,提供故障定位建议。

关键词 RDMA 状态感知 网络度量 Sketch 算法

中图分类号 TP3 **文献标志码** A DOI: 10.3969/j.issn.1000-386x.2022.02.022

RDMA HIGH SPEED NETWORK STATE AWARENESS AND MEASUREMENT METRICS SYSTEM

Xu Jiawei Yan Ming Wu Jie
(*School of Computer Science, Fudan University, Shanghai 200433, China*)
(*The Ministry of Education Network Information Security Audit and Monitoring Engineering Research Center, Fudan University, Shanghai 200433, China*)

Abstract With the explosive growth of data volume carried by data center network, more and more frameworks, models and applications choose to use RDMA technology to accelerate network transmission. RoCE is the implementation of RDMA technology in Ethernet, but there is few effective state awareness and measurement systems for RoCE network, which cannot fully show the state of RDMA network. In view of this situation, this paper proposes a real-time multi-dimensional and multi-level state awareness and measurement metric system for RoCE. It adopted the bypass distributed traffic capture method and used the sketch algorithm to measure the RDMA network state in all aspects. The system is easy to deploy and low cost, with high scalability and flexibility. The experimental results show that the system can objectively reflect the state of RoCE network with low error and provide fault location suggestions.

Keywords RDMA State awareness Network measurement Sketch algorithm

0 引 言

近几年来,数据呈爆炸式增长。根据国际超级计算机 500 强排名的数据,基于 TCP/IP 的传统以太网是世界排名前 500 中最流行的互联方式,随着 40 Gbit/s 乃至 100 Gbit/s 网卡的出现,远程直接内存访问(Remote direct memory access,RDMA)技术已经在世界顶级超级计算领域独领风骚[1],成为目前主流的高性能计算机互连技术之一。RDMA 技术将数据直接从一台计算机的内存传输到另一台计算机,它核心在于将网络层和传输层下移到服务器的硬件网卡

收稿日期:2019-12-23。国家重点研发计划"试验场项目"(2017YFB0803203)。徐佳玮,硕士生,主研领域:计算机网络与分布式系统。严明,工程师。吴杰,研究员。

中，使得数据报文在网卡上完成四层解析后直接到达应用层软件而无须 CPU 的干预。与传统基于 TCP/IP 协议的网络相比，RDMA 网络通过将大多网络功能卸载到物理网卡上、绕开操作系统内核、零拷贝技术实现在提高吞吐量的同时降低时延和 CPU 占用率。

RDMA 技术最早出现在 IB(Infiniband) 网络中，用于高性能计算集群的互联，后来业界厂商把 RDMA 移植到传统以太网上，推动了 RDMA 技术的普及，这样 RDMA 得以部署在目前使用最广泛的数据中心网络上。在以太网上 RDMA 又根据协议栈的不同，分为 RoCE(RDMA over Converged Ethernet) 和 iWARP(RDMA over TCP/IP) 两种技术。根据相关的性能测试，RoCE 相对于 iWARP 吞吐量更高、时延更低[2]。RoCE 分为 RoCE v1 版本和 RoCE v2 版本，RoCE v2 相对于 v1 来说支持了 IP 路由，因此现在 RoCE 网卡大多使用 RoCE v2 版本。随着数据中心网络转发数据的增大，许多数据中心都在利用 RoCE 来加速网络传输。现在许多的框架、人工智能分布式模型训练场景、主流的数据库系统、并行存储文件系统的应用中，也广泛使用了 RDMA 技术[3-4]。另外，随着应用部署的复杂性增大，当出现故障时，非硬件故障性的网络问题不容易轻易复现，因此数据中心非常需要对 RoCE 网络进行实时度量。

然而，目前业界在针对 RoCE 网络的状态感知监控方案和 TCP 网络相比还不成熟，度量指标体系不够完善。首先，现有对 RoCE 网络的度量更多地是以依靠在本地网卡软件抓包之后离线分析或用昂贵的硬件设备分析为主，缺少针对 RoCE 深入到应用层业务流的基于软件实时状态感知系统。其次，从度量指标而言，目前 RDMA 还没有成熟的多维度、多层次、完善的指标体系，RoCE 厂商只给出了系统上硬件统计故障诊断信息，从单节点角度，凭借现有的网卡错误项的累计值不足以了解 RDMA 网络状态和定位问题。另外，根据实验测试结果，采用本地抓包之后离线分析的方式，会对网卡的最大传输性能造成一定损耗。如图 1 的测试结果所示，若在服务器节点本地用 tcpdump 之类的开源工具进行流量捕获从而离线分析的方法，随着消息长度不断增大，当消息大小在 256 字节以上时，本地抓包使得单网卡的最大传输性能下降了 10% ~47%。且对于多节点的离线分析，需要汇总分析结果，二次分析的分析工作复杂。

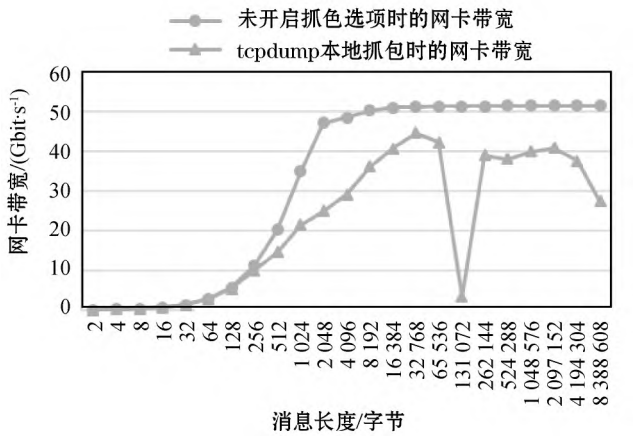

图 1　未开启抓包选项和 tcpdump 抓包时的网卡带宽对比

传统以太网对 TCP 流的分析可以帮助数据中心网络管理员优化网络管理，类似地，对 RoCE 网络进行类似业务流的分析也可以对网络中系统的服务调用和数据传输的流量分布有进一步的认识。随着 RoCE 的普及，对数据包的进一步分析是很有必要的。对传输类型以及基于流等方面的统计度量，有利于进一步研究流控机制、拥塞控制、优化资源调度，也有助于数据中心的网络管理员了解网络情况，进行故障定位和网络规划，有助于指导应用服务的研发人员优化和加速应用。

针对上述问题，本文设计了一种基于软件的 RDMA网络状态感知与度量系统并针对 RoCE 网络进行实现和实验评估。本文的主要贡献如下:

(1) 提出了一个多角度、多维度、多层次的 RoCE 网络指标度量体系，全方位分析网络状态，弥补了现有工作只有硬件统计的错误累积数据、缺失应用层和业务流方面指标而无法全方位感知网络状态的不足;

(2) 设计了一种基于软件的 RDMA 实时网络状态感知和度量系统，采取旁路分布式捕获流量的方式并同时采用 Sketch 流抽样算法对 RDMA 的业务流进一步分析，成本低，易于部署，可扩展性和灵活性高，且对原传输节点的影响较小;

(3) 针对 RoCE 网络实现并模拟网络场景进行实验评估，实验结果显示，本文系统能够在较低的误差范围对 RoCE 网络进行客观的度量，从多角度展现网络状态和问题，能够对数据中心的故障定位和应用服务调度提供一定指导。

1　相关工作

目前工业界和学术界针对 RDMA 的度量主要有三种类型，分别是基于硬件的流量捕获抓包方式、基于软件的流量捕获离线分析方式及针对链路的带宽时延

基准性能测试工具。

文献[5]介绍了 ibdump，它是 Mellanox 公司官方提供的 OFED 驱动的一个抓包组件，其缺陷是当流量速率很高时会丢包，只适用于 Mellanox 的硬件设备，只能作为软件运行在传输节点上，它的最大捕获能力依赖于主机上的 RAM 或者磁盘空间，且随着流量增大会发生无意识无规律的丢包。另外，根据 Mellanox 最新的 OFED 驱动文档，ibdump 仅支持 Connect-X3 和 Connect-X3 Pro[6]，不支持之后的网卡版本。ibdump 只具有抓包功能，而不具有任何流量分析的功能。

文献[7]的 CatC 是 LecCroy 公司的一个硬件分析器。它在部署时必须串联在一条 IB 链路中，部署繁琐，价格昂贵，并且它抓包存储下来的. ibt 文件只能通过本公司的专用 IBTracer 软件打开，此外 CatC 只适用于 SDR（8 Gbit/s）的速率且只有 2 GB 的存储容量。

文献[8]的 tcpdump 是一个常用的流式网络数据采集分析工具，从 OFED3.2 版本、Connect-X4 网卡开始，配合 libpcap 1.9.0-4p150.76.3[9] 及以上版本，tcpdump 可以支持捕获 RDMA 数据包[10]，它对截获的数据并没有进行彻底解码，更适合使用简单的过滤规则保存到文件中，然后再使用其他程序进行离线解码分析。tcpdump 在大流量下受性能和存储容量的影响不适合长期在线实时监测。

文献[11]的 Perftest 和文献[12]的 qperf 是用于各种 RDMA 通讯类型的性能测试应用程序。Perftest 是 Mellanox OFED 驱动中的基准测试程序包，可以用于调优和功能测试。qperf 由 Linux 操作系统默认自带，以用来测试两个节点之间的带宽和延迟，可以用来测试 RDMA 传输的指标。它们都是基于单点的链路角度的测量带宽时延的工具，不涉及对业务数据流的度量。

现有关于 RoCE 度量的研究工作和工具，在基于硬件的流量捕获方面，虽然在性能和精度方面较有优势，但是部署时需要在链路中部署专用硬件，成本较高，灵活性不强；在基于软件的 RoCE 流量度量方面，厂商自产的抓包工具只有抓包功能，开源的抓包分析工具对业务维度方面没有进一步的统计，只有简单的带宽时延测试，缺乏业务数据的分析。并且，基于软件的工具在传输节点上捕获流量时需要开启 RoCE 网卡的 sniffer 模式，这会带来性能损失[13]，影响原本的传输性能。另外，在 RDMA 度量体系来讲，RoCE 厂商除了一些基于硬件上统计信息外，并不关注传输内容和事务本身，RDMA 还没有完善的度量指标体系。

2 RDMA 度量指标体系设计

度量指标体系的层次设计以面向应用业务需求为目的，主要关注应用层的指标，用以指导应用加速优化和网络规划调度，辅以部分网络层面指标和硬件故障统计信息来展示 RoCE 网络状态，用以指导网络问题和故障定位。

原有厂商提供的硬件统计指标只能从单个物理主机节点角度获得某个 RDMA 网卡一些错误信息总体的累计统计值，未细分 RDMA 操作和连接。而当需要进行网络状态感知、服务故障定位，对采用 RDMA 加速的服务进行优化调度时，单从整体统计值上难以判定具体问题，需要分析 RDMA 数据包协议内容，结合应用层面的指标一起联合分析。

如图 2 所示，度量指标体系总体上从网络状态感知和业务应用状态感知两个大层面进行划分。网络状态感知层包括针对节点可用性、连接可用性、网络负载状态。业务状态感知层包括针对 RDMA 操作相关的指标和基于业务流的分析指标。各层次之后从表征的网络模型角度再向下细分，分为链路层、系统层、网络层、应用层。所有单项指标从指标意义角度，分为运行状态指标和负指标。运行状态指标指的是除负指标这类关于软硬件错误和故障信息的指标的其他指标。负指标根据方向的不同，又分为作为发送端检测到的和作为接收端检测到的两个角度。

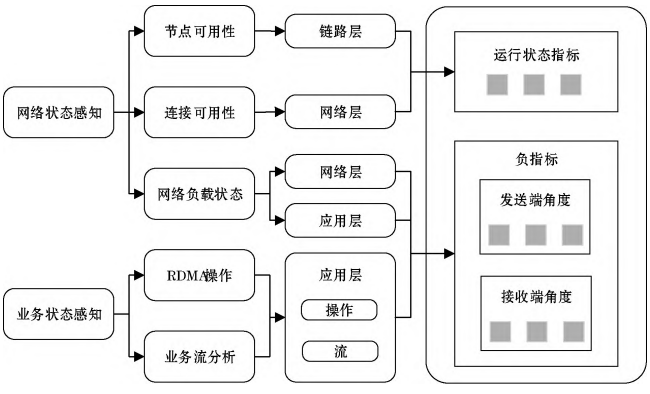

图 2 度量指标体系层次图

2.1 节点可用性指标

节点可用性考虑的是针对网络和服务性能下降时，关于 RDMA 节点方面的状态感知。

（1）指标内容。节点可用性指标内容如表 1 所示。

表 1　节点可用性指标

网络层次	运行状态指标	负指标	
		发送端角度	接收端角度
链路层	RDMA 端口状态 phys_state/state	丢弃发送速率(pps)：因端口 down 或者拥塞导致的被丢弃的出口数据包数量/s port_xmit_discards	/
	/	溢出发送速率(pps)：QP 缺乏 WQE 导致的丢包数量/s out_of_buffer	/

(2) 指标来源。链路层指标通过读取 Linux 系统下的/sys/class/infiniband/的统计文件获取后计算得出[14]，连同系统资源统一在本地收集。

2.2　连接可用性指标

连接可用性指标从网络层数据包角度表征网络状态。

(1) 指标内容。连接可用性指标内容如表 2 所示。

表 2　连接可用性指标

网络层次	运行状态指标	负指标	
		发送端角度	接收端角度
网络层	/	发送 ACK 超时数(个/s)：发送端的 Ack 超时错误的次数/s local_ack_timeout_err	PSN 错误数(个/s)：PSN 不符合预期 PSN 的错误数量/s implied_nak_seq_err
		发送等待时间/usec port_xmit_wait	

(2) 指标来源。负指标通过读取 Linux 系统下的/sys/class/infiniband/的统计文件获取后计算得出。

2.3　网络负载指标

网络负载指标主要针对网络规划和网络拥塞发现两个角度。

(1) 指标内容。网络负载指标内容见表 3。

表 3　网络负载指标

网络层次	运行状态指标	负指标	
		发送端角度	接收端角度
网络层	TCP-RTT/usec：RDMA 传输建立连接时的 TCP RTT	发送拥塞速率/pps：Notification Point 发送的 CNP(Congestion Notification Packet)包数量/snp_cnp_sent	接收失序速率/pps：接收到的失序包的数量/s out_of_sequence
	网卡接收速率(bps/pps)：port_rcv_data、port_rcv_packets	转发失败发送速率/pps：无法转发而被丢弃的数据包数量/s port_rcv_switch_relay_errors	单位 NAK 错误数/pps：收到 NAK 序列号错误的数据包数量/s packet_seq_err
	网卡发送速率(bps/pps)：port_xmit_data、port_xmit_packets	/	单位 RoCE-ICRC 错误数/pps：发生 ICRC 错误的 RoCE 数据包的数量/s rx_icrc_encapsulated
	网卡多播速率/pps：多播数据包数量 port_multicast_rcv_packets	/	接收错包速率/pps：收到错包的数量/s port_rcv_errors
	网卡单播速率/pps：单播数据包数量 port_unicast_rcv_packets	/	ECN 接收速率/pps：接收的 ECN 信号的数量/s np_ecn_marked_roce_packets
应用层	RDMA 操作时延/usec	/	/

尽管传统 RDMA 的应用场景多以高速传输为主，但随着 RDMA 技术在企业应用服务中的应用，RDMA 调用前的连接建立时间对于服务吞吐量也变得尤为重要，因此选用建立 RDMA 连接的 TCP RTT 作为指标，计算请求端进行握手时发出 SYN 包到发送 ACK 的多个连接的平均值作为该连接的 RTT。

(2) 指标来源。RTT 通过度量系统进行在线实时分析计算得出，RDMA 操作时延通过在节点发送探测包计算得出。其余指标通过读取 Linux 系统下的/sys/class/infiniband/的统计文件获取后计算得出。

2.4　RDMA 操作相关流量指标

RDMA 操作相关指标旨在发现网络中各种服务调用中 RDMA 操作类型的分布，从而了解网络中真正关键操作，发现性能瓶颈，指导基于 RDMA 的应用的优化方向。RDMA 各 Opcode 类型流量速率、RDMA 消息吞吐量、Queue pairs 的消耗量可以为研究时延敏感型、吞吐量敏感型、带宽敏感型的应用之间的互相竞争分

析提供依据。

(1) 指标内容。RDMA 操作相关指标内容见表 4。

表 4 RDMA 操作相关指标

网络层次	运行状态指标	负指标	
		发送端角度	接收端角度
应用层	单位接收 atomic 请求速率(个/s):关联 QP 接收到的 atomic 请求的数量/s rx_atomic_requests	单位请求完成队列失败数(个/s):请求端检测到 CQE 错误的次数/s req_cqe_error	单位被动重传数(次/s):接收的重传的请求数量/s duplicate_request
	单位 DCT 接收请求数(个/s)关联 DCT(Dynamically Connected Transport)接收到的连接请求数/s rx_atomic_requests	单位请求完成队列 flush 失败数(个/s):请求端检测到 CQE 发生 flush 错误的次数/s req_cqe_flush_error	单位响应完成队列失败数(次/s):响应端检测到 CQE 错误的次数/s resp_cqe_error
	READ 请求速率(次/s):关联 QP 接收到的 READ 请求的数量/s rx_read_requests	单位远程访问错误数(个/s):请求端检测到的远程访问错误的次数/s req_remote_access_errors	单位请求完成队列 flush 失败数(个/s):响应端检测到 CQE 发生 flush 错误的次数/s resp_cqe_flush_error
	WRITE 请求速率(次/s)关联 QP 接收到的 WRITE 请求的数量/s rx_write_requests	非法请求错误数(个/s):请求端检测到远程不合法请求错误的个数/s req_remote_invalid_request	单位长度错误数(次/s):响应端检测到本地长度错误的次数/s resq_local_length_error
	opcode* 19 种的类型流量速率/(bit·s^{-1})[15](SEND _ FIRSTSEND _ MIDDLESEND_LAST、SEND_LAST_WITH_IMMEDIATE、SEND_ONLY、SEND_ONLY_WITH_IMMEDIATE、RDMA_WRITE_FIRST、RDMA_WRITE_MIDDLE、RDMA_WRITE_LAST、RDMA _ WRITE _ LAST _ WITH _ IMMEDIATE、RDMA_WRITE_ONLY、RDMA_WRITE_ONLY_WITH_IMMEDIATE、RDMA _ READ _ REQUEST、RDMA_READ_RESPONSE_FIRST、RDMA _ READ _ RESPONSE_LAST、RDMA_READ_RESPONSE_ONLY、ACKNOWLEDGE、ATOMIC_ACKNOWLEDGE、COMPARE_SWAP、FETCH_ADD)	/	单位接收非法请求次数(次/s):响应端检测到远程不合法请求错误的个数/s resp_remote_access_errors
		/	RNR NAK 接收速率/pps:收到的 RNR NAK 数据包的数量/s rnr_nak_retry_err
		/	CNP 处理速率/pps:已处理的 CNP 数据包的数量/s rp_cnp_handled
		/	CNP 忽略数(次/s):HCA 反应点接收且忽略的 CNP 数据包的数量/s rp_cnp_ignored
	RDMA 消息吞吐量(个/s)	/	/

(2) 指标来源。负指标通过读取 Linux 系统下的/sys/class/infiniband/的统计文件获取。

opcode 流量速率指标通过度量系统 Sketch 算法进行在线实时分析计算得出。RDMA 消息吞吐量通过系统解析 opcode 中的“LAST”来标记一条消息的结束,从而统计得出。其余指标通过读取 Linux 系统下的/sys/class/infiniband/的统计文件获取后计算得出。

本文目前针对 RoCE 使用最广泛的 Mellanox 的 RoCE v2 协议,其支持 ConnectX-3 Pro 及以上版本的网卡。

如图 3 所示,RoCE 在以太网数据包中封装了 IB。RoCE v2(Layer 3)运作在 UDP/IPv4 或 UDP/IPv6 之上,采用专用端口 4791。

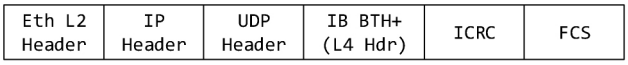

图 3 RoCE v2 报文格式[16]

UDP 协议栈封装了基本数据报头(Base Transport Header,BTH)及 IB 的 payload,其中系统用到的一些 BTH 的字段如下:

① 操作码(8 bits):该字段表示 IBA 数据包的类型。例如 CNP 数据包的操作码是 0x81(10000001b),Reliable Connection(RC)-SEND First 的操作码是 0x00。该字段可以从数据包角度将 RDMA 的操作类型的统计细化到消息和 opcode 层面。

② Solicited Event(SE)(1 bit):该字段表示 responder方是否应该产生一个事件。

③ Pad Count(PadCnt)(2 bits):该字段表示将多少额外字节添加到有效负载以对齐 4 字节边界。

④ 传输层报头版本(TVer)(4 bits):该字段表示 IBA 传输层报头的版本。

⑤ 目的 Queue Pair(DestQP)(24 bits):该字段表示了目的方的 Work Queue Pair(QP)的序号,例如 0x0000d2 = 210。

⑥ 确认请求(A)(1 bit):该字段用来表示需要 responder回复一次确认。

2.5 业务流指标

网络流量测量除了捕获之外需要对流量进一步的解析和分析,从而掌握网络运行状态,提取行为特征。

Queue pairs 可以唯一表示 RDMA 操作,因此定义 RDMA 的流是同一个 Queue pairs 的包序列号连续的所有 RDMA 消息,即 SrcQP + DstQP 或源 IP + 目的 IP + UDP 源端口的消息。RDMA 流的大小的定义是一条流在时间窗口内的 messages 的字节长度总和。在 TCP 网络中,基于流的分析是很有意义的,本文借鉴 TCP 网络中的定义,对 RDMA 流的大小超过配置文件

中设定的阈值的 RDMA 流称为大流，小于阈值的为小流。大流一般对于带宽较为敏感，小流一般对于时延较为敏感。

RDMA 流维度的相关指标包括 RDMA 流总个数、RDMA 流速率大小、RDMA 大流个数、流抖动率、小流时延。

3　RDMA 状态感知度量系统总体架构

系统框架如图 4 所示，系统总体分为度量系统和 Agent。度量系统运行在网络中独立的一台服务器上，网络流量由交换机镜像口镜像，可经过分流器分流，连接到多个度量系统所在的服务器上，捕获流量的网卡是以太网网卡。部署环境要求在被镜像出的流量在进入服务器之前的链路不发生丢包且不存在重复流量。

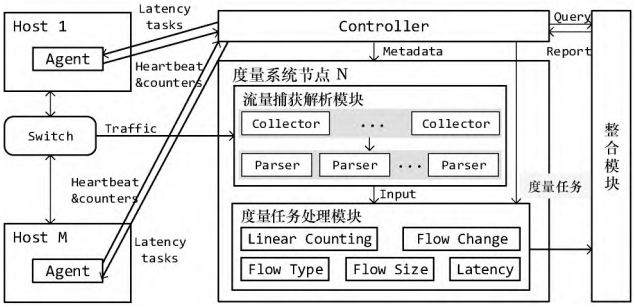

图 4　系统架构图

度量系统主要负责流量的旁路捕获和分析。Agent作为度量系统的信息收集辅助工具运行在每个 RDMA 传输节点上，主要负责监听传输节点的硬件网卡自带的网卡、端口统计，辅助探测链路的时延，并发送给 Controller，对本地节点的影响很小。对于需要测量时延的节点，Controller 将指令发送给其上的 Agent，之后 Agent 发送探测数据包得到节点之间的时延发送给 Controller。

采用旁路分布式全量捕获流量和 Agent 辅助收集基础信息相结合的方式的优点是实现了在高流量场景下，在减轻观察者效应的同时，可以从网络和应用两个角度，兼顾实现多维度指标的收集。

度量系统从架构上分为控制平面和数据平面。

1）控制平面。控制平面是 Controller 控制模块。Controller 模块主要负责配置相关元数据信息、添加度量处理任务、进行相关网络状态的查询操作，它还负责和节点上的 Agent 交互，检测 Agent 的心跳，接收 Agent 上报的节点上的 RDMA 端口、错误信息和时延相关数据。元数据包括服务器网卡 IP 端口、流处理的任务和类型、统计间隔，以及 Agent 采集信息的频率和 Sketch 算法的参数如时间间隔、黑白名单、任务类型、大小流阈值等一系列可配置项。配置文件使用 YAML 格式进行描述。

2）数据平面。数据平面分为流量捕获解析模块、度量任务处理模块、整合模块。

（1）流量捕获解析模块。Collector 从网卡中全量捕获流量，Parser 根据 RDMA 协议格式解析数据包各个字段，提取有效信息。服务器每个网卡对应一个 Collector，之后由多个 Parser 并行进行数据包的解析处理。

（2）度量任务处理模块。根据配置中新建若干度量任务，每个任务都会利用流量捕获解析模块的输出使用不同的 Sketch 算法进行统计，将统计结果输入到整合模块。

（3）整合模块。整合模块汇总整合各个分布式度量节点处理的结果，对于某个指标，累计分流到的节点的计算数值，并提供给 Controller 查询统计值的接口或者输出成离线文件。

4　系统主要功能具体实现

4.1　流量捕获方法

基于硬件的流量捕获方式有精度和性能的优势，但是不易部署且价格昂贵。本文采用软件旁路方式进行分布式流量捕获，采用软件方式比较轻量，可以根据配置和网络情况指定过滤规则，以插件方式添加流量分析处理任务，使得指标显示可组合，分析更加灵活。

流量捕获解析模块中，Collector 使用零拷贝技术，根据配置文件全量捕获设定的时间窗口的数据包。度量系统根据流量大小和服务器资源情况可以是一个或多个，当需要处理的流量很大时，可将总流量分成交换机的多个端口镜像再经过分流器基于 IP 规则、方向、RoCE 中的序列号字段过滤出分散的无重复的流量，再输入到分布式度量服务器的多个网卡中并行处理。采用多网卡分布式抓包后整合的方式实时分析流量，整合模块汇总显示各网卡的统计结果。

4.2　RDMA 流维度的指标度量方法

在数据中心中，更关注流量的分布和趋势，没有必要使用 hash table 类型的计数器统计获取非常精确的数值。采用基于概率统计理论的 Sketch 算法统计 RDMA流维度指标虽然会产生一定误差，但是其误差在可以接受的范围之内，足以获取需要的流量分布和趋势特征[17]。此外，sketch 的内存占用更小，降低了设备的成本。

如图 5 所示，度量任务类型根据 Controller 配置

文件下发。流量捕获分析模块采用纯软件方法解析出发送端、接收端、操作码、Queue Pair、数据包大小之后，判断数据包所属的流是否是新流，更新时间窗口内的流总个数和流大小。对于高于一定阈值的大流来讲，判断大流的抖动情况；对于小于一定阈值的流来讲，进一步向 Agent 发送信号，示意 Agent 定期发送时延检测数据包，最后 Agent 将结果返回给 Controller。

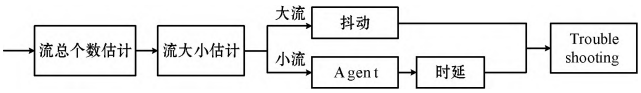

图 5 RDMA 流维度度量流程

(1) RDMA 流总个数估计。流总数估计任务使用 Linear Counting(LC) 算法[18] 估计一个时间窗口内不同的流的个数。LC 算法是一种计数估计算法，假设一个哈希函数结果服从均匀分布，结果空间为{0，1，…，$m-1$}的 m 个值。再有一个长度为 m 的 bitmap，每个比特都是初始化为 0。对流式数据里每个元素，也就是 RDMA 流的标识，LC 算法先通过哈希函数将标识映射到 bitmap 的某个比特 x 上，如果第 x 个比特为 0，那么将第 x 个比特设置为 1。查询时，根据 1 的个数和比例得到关于 RDMA 流的总数的估计。

(2) RDMA 流大小估计、RDMA opcode 大小估计。流大小估计任务和 RDMA opcode 大小估计任务使用 Count-Min(CM) 算法[19] 获得任意流在某时间窗口内的大小估计值，即 RDMA 流速率，从而在下一阶段针对不同类型的流进行进一步的分析。CM 算法是一种概率数据结构，维护一个长度为 i 的数组，每个数组对应 j 个计数器和一个对应的哈希函数，初始化每个计数器为 0。面对流式数据里每个新的事件元素，计算每个元素在每个数组中根据哈希得到的值，作为数组的位置索引，然后将数组对应位置索引下的计数器加一。当需要查询某个元素时，返回这个元素在不同数组中计数值的最小值。

根据 Controller 的配置，判定 RDMA 流大小大于设定阈值的流为大流，小于设定阈值的流为小流，分情况进行下一步度量。

(3) 大流抖动检测。抖动检测是针对大流，估计在两个相邻的时间窗口内，流的大小变化是否超过一定的阈值。使用与 Count-Min 类似的数据结构，通过比较两个时间窗口的值的偏差和上一个时间窗口的值的比值，检测流量大小是否发生抖动。

(4) 小流检测。对于网络中大小超过一定阈值的小流，进一步探测传输节点之间的时延。Controller 查询当前时间窗口流大小前 p 小的流对应的传输节点，将信号发送给节点上的 Agent，每个 Agent 都可以当做 Server 和 Client。在下一个时间窗口内，Agent 以一定频率连续发送 10 个时延探测包。在时间窗口结束后，将时延探测平均值返回给 Controller。

5 实 验

5.1 物理实验环境

本文使用了四台服务器与一台交换机，实验环境如图 6 所示，其中三台服务器用于传输数据且部署 Agent，另一台服务器用作运行度量系统。

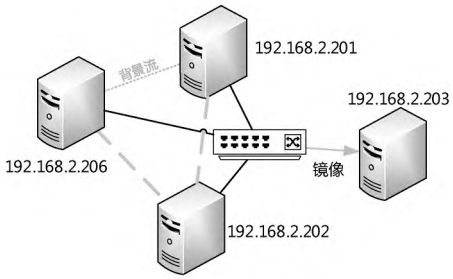

图 6 实验环境拓扑图

四台服务器使用 RoCE 100G ConnectX-5 网卡，服务器系统上安装的网卡 OFED 驱动版本为 MLNX_OFED_LINUX-4.0-2.0.0.1，交换机型号为 Mellanox SN2100，服务器具体规格如表 5、表 6 所示。

表 5 服务器操作系统和内核版本

服务器	操作系统	内核版本
192.168.2.201	CentOS 7.6.1810	3.10.0-957.el7.x86_64
192.168.2.202	CentOS 7.4.1708	3.10.0-693.el7.x86_64
192.168.2.203	14.04.1-Ubuntu	4.4.0-142-generic
192.168.2.206	14.04.1-Ubuntu	4.4.0-142-generic

表 6 服务器 CPU 和内存

服务器	CPU 型号	内存
192.168.2.201	Intel(R) Xeon(R) CPU E5-2643 v4 @ 3.40 GHz	16 GB
192.168.2.202	Intel(R) Xeon(R) CPU E5-2687W v4 @ 3.00 GHz	16 GB
192.168.2.203	Intel(R) Xeon(R) CPU E5-2687W v4 @ 3.00 GHz	32 GB
192.168.2.206	Intel(R) Xeon(R) CPU E5-2620 v4 @ 2.10 GHz	32 GB

传输节点之间使用官方的 Perftest 基准性能测试工具，以及用 RDMA ibverbs 标准库函数编写的程序，

模拟客户端和服务端发送流量。

5.2　指标验证

本节利用状态感知度量系统指标讨论不同网络状态下的指标特征变化情况。实验过程中，单节点度量系统服务器 CPU 占用率平均约为 64%，Agent 对服务器的 CPU 占用率基本低于 2%，对原传输节点的资源影响非常小。

(1) RDMA RC-Write/Send/Read 操作时延随流量变化情况。图 7 显示了单流情况下基于 RDMA 可靠传输 Write/Send/Read 操作随流量增长的变化情况，Write 操作的操作时延一般性低于 Send 和 Read 操作，符合 RDMA 的特征，并且度量系统分析的结果和通过 tcpdump 抓包后离线分析呈现同样的结果。

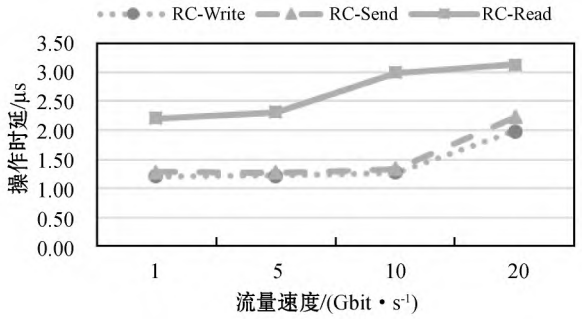

图 7　负指标数值的增长情况

(2) 建立连接时 TCP RTT 随流量变化情况。如图 8 所示，实验测量了两个传输节点单流情况下随着流量不断增大，再次建立 RDMA 连接时的 TCP RTT 平均时间。当流量在 10 Gbit/s 以上时，建立连接时的网络时延有明显增大。度量系统分析的结果和通过 tcpdump 抓包后离线分析呈现同样的结果。

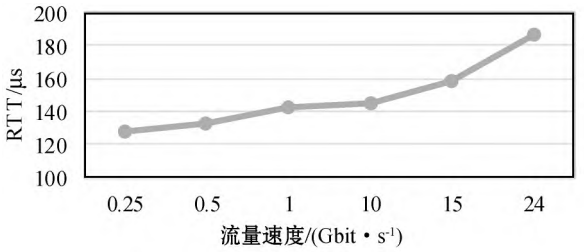

图 8　建立连接时 TCP RTT 随流量变化

(3) RoCE 流个数与流大小指标验证。本节测试多节点多流时的流大小和个数分布情况，在 192.168.2.201 服务器和 192.168.2.20 之间产生的 10 条速率小于 5 Mbit/s 的稳定 RDMA 背景流，另外在三台服务器上产生四条速率分别为 1/3/4/6 Gbit/s 的流。多流分布情况如图 9 所示，接近横坐标轴的实线部分的流 flow1 – flow10 是背景流。系统可以客观显示网络中的流分布情况。

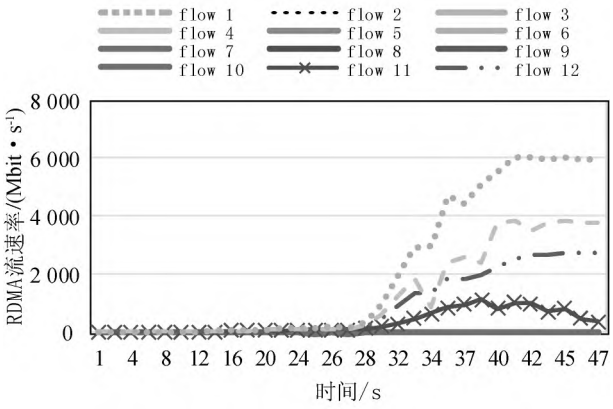

图 9　多流流大小分布情况

(4) RDMA 操作的相互影响。在 192.168.2.206 和 192.168.2.202 之间产生 25 个 QP 用来进行 RC-Send/Write/Read、UD/UC-Send、UC Write 操作传输流量，所有操作同时开始，模拟多操作并存时的流量相互影响情况，同时作为下一节网络拥塞场景模拟的流量来源。

如图 10 所示，在流量传输过程中，Read 操作程序产生访问远程地址错误日志，而状态感知系统可以客观显示由于流之间存在相互影响，Read 操作先占用了大部分资源，导致其他操作在建立连接时略微滞后。

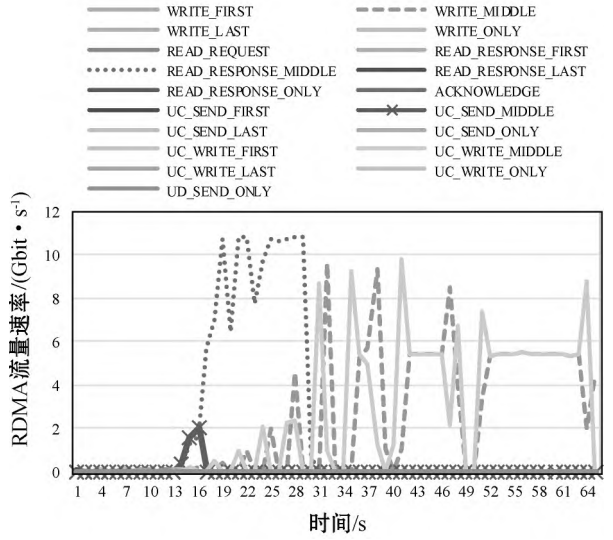

图 10　多流 RDMA 操作的流量大小分布情况

5.3　RDMA 常见场景问题验证

1) 场景一: 当传输过程中接收端程序断开导致连接断开的情况模拟。模拟在进行 RC-Send 传输中接收端程序突然中止的情况，RDMA 操作相关指标变化累计和如图 11 所示。

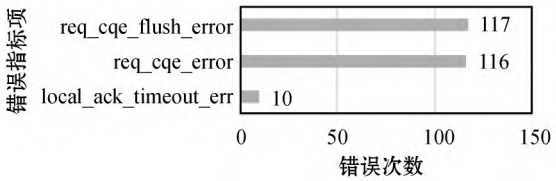

图 11　场景一 RDMA 操作相关指标变化

发送端的 Ack 超时错误的次数、请求端检测到 CQE 错误的次数、请求端检测到 CQE 发生 flush 错误的次数都同时增长。但网卡与链路的状态无任何异常，节点可用性指标、连接可用性指标、网络负载指标正常，排除网络拥塞造成的传输故障，表征了接收端的应用程序存在着故障。

(2) 场景二：网络拥塞模拟。使用如图 10 所示的同样的流量生成方法，模拟单节点发送网络拥塞的场景。如图 12、图 13 所示，当 Read 操作程序产生访问远程地址错误日志时，RDMA 操作时延较正常情况下增加了 9.1% ~87.7%，同时发送端和接收端的网络负载相关指标同时变成非零，并且接收端的 CNP 信号(responder-rp_cnp_handled)数量较其他指标在传输时段的累积和更多，因此可以说明是发送端节点导致的网络拥塞。

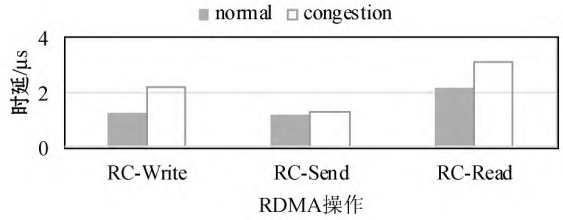

图 12 RDMA RC 操作时延的前后对比

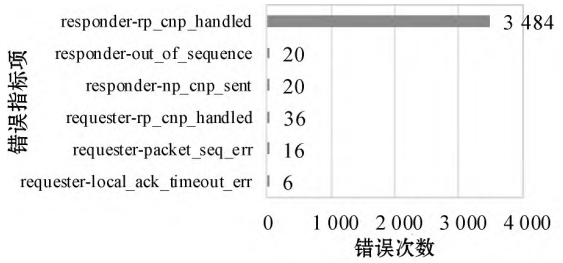

图 13 场景二网络负载相关指标数值在传输阶段的累计

(3) 场景三：大流抖动模拟。针对带宽敏感的应用场景，模拟在大流出现抖动的情况，同样产生如图 9 中的 10 条背景 RDMA 流，另外产生流量趋势随机变化三条高速率的 RDMA 流，设定大流阈值为 5 Gbit/s，抖动阈值为 2.5 Gbit/s，抖动窗口为 1 秒。如图 14 所示，度量系统可以准确识别出三条流的大小抖动变化。

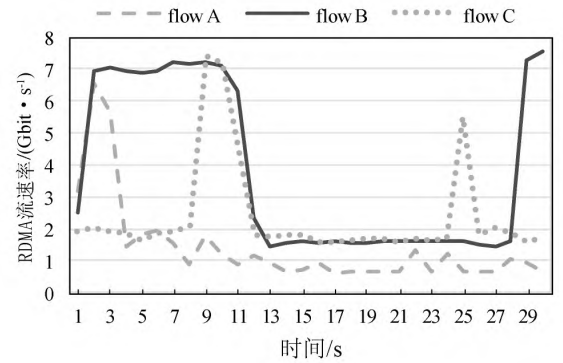

图 14 场景三大流抖动检测

5.4 误差评估

本文选取识别、检索、分类、翻译等领域常用评估指标：平均绝对误差、平均相对误差和准确率。

(1) 平对绝对误差。平均绝对误差是所有流某个指标测量值和实际值偏差的绝对值和实际值的比值的平均。对于流的大小的估计和 Opcode 流量速率估计将使用平均绝对误差来评估误差。

(2) 相对误差。相对误差是绝对误差和测量值的平均值的比值的平均值，通常用绝对值。对于总的流个数的估计使用相对误差来评估误差。

(3) 准确率。准确率是系统正确识别的指标项个数和应当被识别出的指标项个数的比值的平均值。

除从系统上收集的文件数据外，其余所有指标的真实值采用 RoCE 官方的 Perftest 基准性能测试工具以及用 tcpdump 同步抓包离线分析的结果，和系统计算出的各项指标进行误差计算。

(1) RoCE 流个数估计误差。网络中的总的 RoCE 流个数采用相对误差验证，在三个节点之间随时间变化产生共 1 ~72 条流量速率范围在 0.5 Mbit/s ~20 Gbit/s的流，估计值和真实值的相对误差小于 2.7%。

(2) 流大小估计和 Opcode 操作流量估计的误差。流大小分布估计和 RDMA 操作流量估计采用平均绝对误差，流量模拟分别采用 5.2 节(3)、(4)中的流量产生方法，测试时长为 2 分钟，每两秒计算一次，两者的平均绝对误差分别为 4% 和 11.3%。

(3) 大流和流抖动识别误差。产生 5 条流量速率小于 0.5 Gbit/s 背景 RDMA 流，5 条速率在 1 ~5 Gbit/s 的 RC-send RDMA 流，与离线文件对比，大流阈值设定梯度为 1.5/2.5/3.5 Gbit/s，抖动阈值设置为 2 Gbit/s，流量速率大小分别变化 4 次，测的大流和抖动识别准确率平均值为 95%。

经过误差分析，该状态感知度量系统可以客观地反映网络状态的变化趋势。

6 结 语

本文设计、实验并评估了一个可扩展、低成本、面向多点网络的 RDMA 网络在线实时状态感知度量系统，设计了多维度多层次的 RoCE 网络度量指标体系，在 RoCE 网络中针对不同类型和大小的流量对 RDMA 的特征数据进行估计，由此给出一些可供参考的故障诊断意见。结果显示，系统可以在较低误差下多方位展现网络状态。

下一步工作将细化 RDMA 的操作，将更多的指标

变化的含义和网络错误匹配，另外考虑增加系统对 iWARP 和 IB 网络场景下的适用支持。

参 考 文 献

[1] TOP500 Meanderings: InfiniBand fends off supercomputing challengers [EB/OL]. (2017-12-12) [2019-12-23]. https://www.top500.org/news/top500-meanderings-infiniband-fends-off-supercomputing-challengers/.

[2] Mittal R, Shpiner A, Panda A, et al. Revisiting network support for RDMA [C]//Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication. ACM, 2018: 313 - 326.

[3] Rdma-based apache Hadoop [EB/OL]. (2018-01-25) [2019-12-23]. http://hibd.cse.ohio-state.edu/.

[4] Rdma-based apache Spark [EB/OL]. (2018-01-25) [2019-12-23]. http://hibd.cse.ohio-state.edu/.

[5] Monitoring and debug tool [EB/OL]. (2014) [2019-12-23]. https://www.mellanox.com/page/products_dyn?product_family=110&menu_section=34&ssn=a9oko89dmm9k1aqnfr4v0dfip0.

[6] InfiniBand fabric utilities [EB/OL]. (2019-07-31) [2019-12-23]. https://docs.mellanox.com/display/MLNXOFEDv461000/InfiniBand+Fabric+Utilities.

[7] The CATC trace: Expert protocol analysis software [EB/OL]. (2003-04) [2019-12-23]. http://cdn.teledynelecroy.com/files/whitepapers/wpcatctrace_0403.pdf.

[8] Tcpdump [EB/OL]. (2019-09-30) [2019-12-23]. https://www.tcpdump.org/.

[9] libpcap1-1.9.0-lp150.76.3.x86_64.rpm [EB/OL]. (2018-09-24) [2019-12-23]. https://opensuse.pkgs.org/15.0/network-utilities/libpcap1-1.9.0-lp150.76.3.x86_64.rpm.html.

[10] How-To dump RDMA traffic using the inbox tcpdump tool (ConnectX-4) [EB/OL]. (2018-12-05) [2019-12-23]. https://mymellanox.force.com/mellanoxcommunity/s/article/how-to-dump-rdma-traffic-using-the-inbox-tcpdump-tool-connectx-4-x.

[11] Perftest package [EB/OL]. (2019-02-20) [2019-12-23]. https://community.mellanox.com/s/article/perftest-package.

[12] qperf(1)-Linux man page [EB/OL]. [2019-12-23]. https://linux.die.net/man/1/qperf.

[13] Offloaded Traffic Sniffer [EB/OL]. (2018) [2019-12-23]. https://docs.mellanox.com/display/MLNXOFEDv451010/Offloaded+Traffic+Sniffer.

[14] Understanding mlx5 Linux counters and status parameters [EB/OL]. (2018-12-05) [2019-12-23]. https://community.mellanox.com/s/article/understanding-mlx5-linux-counters-and-status-parameters.

[15] source code of linux/include/rdma/ib_pack.h [EB/OL]. [2019-12-23]. https://code.woboq.org/linux/linux/include/rdma/ib_pack.h.html.

[16] Annex A17: RoCEv2 [EB/OL]. (2014-09-02) [2019-12-23]. https://cw.infinibandta.org/document/dl/7781.

[17] Yang T, Jiang J, Liu P, et al. Elastic sketch: Adaptive and fast network-wide measurements [C]//Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication. ACM, 2018: 561 - 575.

[18] Whang K Y, Vander-Zanden B T, Taylor H M. A linear-time probabilistic counting algorithm for database applications [J]. ACM Transactions on Database Systems, 1990, 15(2): 208 - 229.

[19] Cormode G, Muthukrishnan S. An improved data stream summary: The count-min sketch and its applications [C]//6th Latin American Symposium: Theoretical Informatics. Springer, 2004: 29 - 38.

(上接第 119 页)

[12] Zhang K, He B, Hu J, et al. G-NET: Effective GPU sharing in NFV systems [C]//15th USENIX Symposium on Networked Systems Design and Implementation (NSDI 18), 2018: 187 - 200.

[13] Zheng Z, Bi J, Wang H, et al. Grus: Enabling latency SLOs for GPU-Accelerated NFV systems [C]//2018 IEEE 26th International Conference on Network Protocols (ICNP), 2018.

[14] Bremler-barr A, Harchol Y, Hay D. OpenBox: A software-defined framework for developing, deploying, and managing network functions [C]//2016 ACM SIGCOMM Conference, 2016.

[15] Katsikas G P, Enguehard M, Kuźniar M, et al. SNF: synthesizing high performance NFV service chains [J]. PeerJ Computer Science, 2016, 2: e98.

[16] Barbette T. Architecture for programmable network infrastructure [D]. University of Liege, 2018.

[17] Liu G, Ren Y, Yurchenko M, et al. Microboxes: High performance NFV with customizable, asynchronous TCP stacks and dynamic subscriptions [C]//2018 Conference of the ACM Special Interest Group, 2018.

[18] Lakshman T V, Stiliadis D. High-speed policy-based packet forwarding using efficient multi-dimensional range matching [J]. ACM SIGCOMM Computer Communication Review, 1998, 28(4): 203 - 214.

[19] Aho A V, Corasick M J. Efficient string matching: An aid to bibliographic search [J]. Communications of the ACM, 1975, 18(6): 333 - 340.