Appeared as a Poster at USENIX FAST 2024

Rack-Scale Memory Disaggregation over Ethernet

Weigao Su
Purdue University

Vishal Shrivastav
Purdue University

1 Motivation

Rack-scale memory disaggregation promises several benefits, including high compute density, fine-grained resource pooling and provisioning, seamless resource scaling, and independent evolution of resources. Existing network stacks for rack-scale memory disaggregation, such as TCP/IP and RDMA (RoCEv2), are built on top of Ethernet Media Access Control (MAC) layer. Unfortunately, this results in several limitations, both in terms of latency and bandwidth utilization, for memory traffic.

- **Limitation 1:** *Minimum frame size overhead.* Ethernet MAC imposes a minimum frame size of 64 B. This may result in extremely poor bandwidth utilization for inherently small memory flows, which may be much smaller than 64 B.
- **Limitation 2:** *Inter-frame gap (IFG) overhead.* IEEE 802.3ae requires a minimum gap of 96 bits (called idle bits) between two consecutive frames. This results in a significant bandwidth overhead for small frames—16% overhead for 64 B frames.
- **Limitation 3:** *No intra-frame preemption.* When memory and traditional traffic coexist, interference can be reduced using priority classes. However, since a frame's transmission at the MAC layer can't be preempted, this results in significant overhead for memory traffic.
- **Limitation 4:** *Layer 2 switching overhead.* Each Ethernet frame will be processed and forwarded by a Top-of-Rack switch, where it goes through several modules [4], including a parser, one or more match-action stages for table look-ups, and a packet manager. The overall latency can be several 100s of ns for state-of-the-art switches.
- **Limitation 5:** *Transport layer overhead.* Existing transport layer [13, 19] embeds complex reliability, congestion and flow control protocols on top of Ethernet to handle in-network queuing and frame losses. They inevitably add latency to data path to/from packet headers, which requires header parsing, header encapsulation/decapsulation, per-flow state updates and look ups.
- **Limitation 6:** *Queuing delay at the switch.* Popular reactive congestion control protocols [1–3, 9, 12] lead to significant network queuing at high loads. Frame loss due to queuing can cause high latency for small memory flows, as they may not trigger fast retransmission but timeouts which are set conservatively at several μs [1, 3, 8]. While Priority Flow Control (PFC) can reduce retransmission overhead, it doesn't alleviate queuing delay. Recent proactive congestion control proposals [5–8, 14] schedule flows to avoid queuing, but their decentralized nature can cause scheduling conflicts.

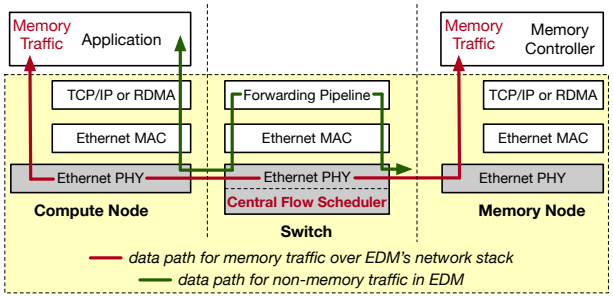

Figure 1: EDM's end-to-end network stack for memory disaggregation over Ethernet.

2 Our Approach and Results

We present **EDM** (**E**thernet **D**isaggregated **M**emory), built around two key design ideas. First, the entire network stack for memory disaggregation is implemented inside the physical (PHY) layer (Figure 1), thus bypassing the overheads of higher layers. We design both a novel PHY layer processing at hosts and forwarding at the switch for memory traffic. Second, EDM presents a *centralized* memory flow scheduler running in the Top-of-Rack switch, that *proactively* avoids congestion, resulting in *zero* network queuing with high bandwidth utilization. The scheduler can make scheduling decisions in only a few nanoseconds using a novel hardware pipeline inspired from prior works [11, 15–18].

Using an FPGA testbed, we show that EDM's network stack only adds ~250 ns over an unloaded network, which is 4×, 9×, and 19× lower than the latency of raw Ethernet, RDMA over converged Ethernet (RoCEv2), and hardware-offloaded TCP/IP network stacks respectively. More importantly, this latency is comparable to a one hop NUMA latency inside a server [10]. Further, rack-scale network simulations suggest that even at high network loads, EDM's latency is within 1.3× of its baseline latency over an unloaded network.

References

[1] ALIZADEH, M., GREENBERG, A., MALTZ, D. A., PADHYE, J., PATEL, P., PRABHAKAR, B., SENGUPTA, S., AND SRIDHARAN, M. *Data Center TCP (DCTCP).* SIGCOMM, 2010.

[2] ALIZADEH, M., KABBANI, A., EDSALL, T., PRABHAKAR, B., VAHDAT, A., AND YASUDA, M. *Less Is More: Trading a Little Bandwidth for Ultra-Low Latency in the Data Center.* NSDI, 2012.

[3] ALIZADEH, M., YANG, S., SHARIF, M., KATTI, S., MCKEOWN, N., PRABHAKAR, B., AND SHENKER, S. *pFabric: Minimal Near-optimal Datacenter Transport.* SIGCOMM, 2013.

[4] BOSSHART, P., GIBB, G., KIM, H.-S., VARGHESE, G., MCKEOWN, N., IZZARD, M., MUJICA, F., AND HOROWITZ, M. *Forwarding Metamorphosis: Fast Programmable Match-Action Processing in Hardware for SDN.* SIGCOMM, 2013.

[5] CAI, Q., ARASHLOO, M. T., AND AGARWAL, R. *dcPIM: Near-optimal Proactive Datacenter Transport.* SIGCOMM, 2022.

[6] CHO, I., JANG, K., AND HAN, D. *Credit-Scheduled Delay-Bounded Congestion Control for Datacenters.* SIGCOMM, 2017.

[7] GAO, P. X., NARAYAN, A., KUMAR, G., AGARWAL, R., RATNASAMY, S., AND SHENKER, S. *PHost: Distributed near-Optimal Datacenter Transport over Commodity Network Fabric.* CoNEXT, 2015.

[8] HANDLEY, M., RAICIU, C., AGACHE, A., VOINESCU, A., MOORE, A., ANTICHI, G., AND WOJCIK, M. *Re-architecting datacenter networks and stacks for low latency and high performance.* SIGCOMM, 2017.

[9] JACOBSON, V. *Congestion avoidance and control.* SIGCOMM, 1988.

[10] KUMAR, A. *The New Intel® Xeon® Processor Scalable Family (Formerly Skylake-SP).* HotChips, 2017.

[11] LEE, K. S., WANG, H., SHRIVASTAV, V., AND WEATHERSPOON, H. *Globally Synchronized Time via Datacenter Networks.* SIGCOMM, 2016.

[12] LI, Y., MIAO, R., LIU, H. H., ZHUANG, Y., FENG, F., TANG, L., CAO, Z., ZHANG, M., KELLY, F., ALIZADEH, M., AND YU, M. *HPCC: High Precision Congestion Control.* SIGCOMM, 2019.

[13] MITTAL, R., LAM, T., DUKKIPATI, N., BLEM, E., WASSEL, H., GHOBADI, M., VAHDAT, A., WANG, Y., WETHERALL, D., AND ZATS, D. *TIMELY: RTT-based Congestion Control for the Datacenter.* SIGCOMM, 2015.

[14] MONTAZERI, B., LI, Y., ALIZADEH, M., AND OUSTERHOUT, J. *Homa: A Receiver-Driven Low-Latency Transport Protocol Using Network Priorities.* SIGCOMM, 2018.

[15] SHRIVASTAV, V. *Fast, Scalable, and Programmable Packet Scheduler in Hardware.* SIGCOMM, 2019.

[16] SHRIVASTAV, V. *Programmable Multi-Dimensional Table Filters for Line Rate Network Functions.* SIGCOMM, 2022.

[17] SHRIVASTAV, V. *Stateful Multi-Pipelined Programmable Switches.* SIGCOMM, 2022.

[18] SHRIVASTAV, V., LEE, K. S., WANG, H., AND WEATHERSPOON, H. *Globally Synchronized Time via Datacenter Networks.* Transactions on Networking, 2019.

[19] ZHU, Y., ERAN, H., FIRESTONE, D., GUO, C., LIPSHTEYN, M., LIRON, Y., PADHYE, J., RAINDEL, S., YAHIA, M. H., AND ZHANG, M. *Congestion Control for Large-Scale RDMA Deployments.* SIGCOMM, 2015.