Understanding RDMA Behavior in NUMA Systems

Jacob Nelson
Computer Science and Engineering Department
Lehigh University
Bethlehem, PA, USA
jjn217@lehigh.edu

Roberto Palmieri
Computer Science and Engineering Department
Lehigh University
Bethlehem, PA, USA
palmieri@lehigh.edu

***Abstract*—Most high performance computing clusters are nowadays composed of large multicore machines that expose Non-Uniform Memory Access (NUMA), and they are interconnected using modern communication paradigms, such as Remote Direct Memory Access (RDMA). In this work we perform a study outlining the performance impact of these two technologies, NUMA and RDMA, when combined. Findings show that system's software architecture should be designed for NUMA and RDMA; otherwise major performance penalties occur.**

***Index Terms*—Non-uniform Memory Access, Remote Direct Memory Access, Distributed Systems**

I. Introduction

High performance distributed systems are commonly developed relying on the message passing programming model, in which data is sent between nodes via messages. Message passing allows for easy interaction and clear abstraction, but performance suffers from required computation by both participants in the communication. In contrast, the shared memory programming model in distributed systems enables programmers to access memory address space that is shared across nodes, thus nodes can access remote memory as if it were local.

These programming models were clearly separated until the adoption of RDMA (Remote Direct Memory Access) motivated a hybrid model combining the strengths of shared memory and message passing [1]. In RDMA, a network interface controller (NIC) communicates directly with the host memory controller, enabling access to remote memory without involving the remote CPU, called *one-sided* interaction. One-sided communication enables algorithmic innovations [1]–[3] and reduces overhead, therefore lower latency and higher throughput.

The advantage of avoiding allocating OS resources to handle remote memory requests can disadvantage performance when machines equipped with RDMA have NUMA (Non-Uniform Memory Access) [4] architectures. NUMA causes memory access latency to vary depending on physical location of the requested memory. Message passing systems avoid costs by pinning data and threads to the same NUMA zone as the NIC [5] and using NUMA-aware data structures for processing [6]. However, one-sided communication conceals memory access patterns from the remote node and hinders mitigating the adverse effects of NUMA.

II. Methodology

Our goal is to isolate performance penalties produced by the interaction between NUMA and RDMA, identify their causes, and deliver guidelines for developing applications that use one-sided transports. Note that any multicore computing node hosting more than one CPU-socket currently deploys NUMA, and that any state-of-the-art high performance distributed testbed leverages RDMA [7], [8]. To the best of our knowledge, our investigation is the first of its kind.

We target applications with concurrent machine-local and RDMA accesses, including distributed graph processing, real-time data analytics, and transactional systems that co-locate serving threads with application threads [2], [5], [9].

In the following we report the highlights of our findings for one-sided RDMA interaction on NUMA machines. NUMA-local refers to resources in the NIC's NUMA zone; NUMA-remote refers to any other NUMA zone. For our tests we use a Mellanox ConnectX-3 network adapter [10] and nodes with two Intel Xeon E5-2650v2 processors totaling 32 logical cores.

- RDMA reads and writes respond differently to NUMA-remote memory. Under no machine-local load, RDMA writes are faster than reads. But, when machine-local write load is high, client read throughput for NUMA-remote memory is up to 3.7x better than writes.
- Local worker thread location matters, even when data is NUMA-local. Compared to no worker NUMA policy, NUMA-local RDMA writes show 3.6x - 5x improvement in throughput with all worker threads running on NUMA-local cores.
- After initialization, shared memory buffer pages needed by RDMA are locked and cannot be moved across NUMA zones; OS cannot enable policy for NUMA access optimizations [11], therefore applications must decide where RDMA-accessed shared memory is allocated a priori.

Our research enables the full exploitation of RDMA with NUMA machines.

Acknowledgment

This material is based upon work supported by the National Science Foundation under Grant No. CNS-1814974.

REFERENCES

[1] M. K. Aguilera, N. Ben-David, I. Calciu, R. Guerraoui, E. Petrank, and S. Toueg, "Passing messages while sharing memory," in *Proceedings of the 2018 ACM Symposium on Principles of Distributed Computing*. ACM, 2018, pp. 51–60.

[2] A. Dragojević, D. Narayanan, O. Hodson, and M. Castro, "Farm: Fast remote memory," in *Proceedings of the 11th USENIX Conference on Networked Systems Design and Implementation*, 2014, pp. 401–414.

[3] H. Chen, R. Chen, X. Wei, J. Shi, Y. Chen, Z. Wang, B. Zang, and H. Guan, "Fast in-memory transaction processing using rdma and htm," *ACM Transactions on Computer Systems (TOCS)*, vol. 35, no. 1, p. 3, 2017.

[4] C. Lameter, "Numa (non-uniform memory access): An overview," *Queue*, vol. 11, no. 7, pp. 40:40–40:51, Jul. 2013.

[5] F. Yang, M. Wu, J. Xue, W. Xiao, Y. Miao, L. Wei, H. Lin, Y. Dai, and L. Zhou, "Gram: Scaling graph computation to the trillions," in *SoCC*. ACM - Association for Computing Machinery, August 2015.

[6] H. Daly, A. Hassan, M. F. Spear, and R. Palmieri, "NUMASK: high performance scalable skip list for NUMA," in *32nd International Symposium on Distributed Computing, DISC 2018, New Orleans, LA, USA, October 15-19, 2018*, ser. LIPIcs, vol. 121, 2018, pp. 18:1–18:19.

[7] R. Ricci, E. Eide, and The CloudLab Team, "Introducing CloudLab: Scientific infrastructure for advancing cloud architectures and applications," *USENIX*, vol. 39, no. 6, Dec. 2014.

[8] J. Mambretti, J. Chen, and F. Yeh, "Next generation clouds, the chameleon cloud testbed, and software defined networking (sdn)," in *Proceedings of the 2015 International Conference on Cloud Computing Research and Innovation (ICCCRI)*, ser. ICCCRI '15. Washington, DC, USA: IEEE Computer Society, 2015, pp. 73–79.

[9] B. Babcock, S. Babu, M. Datar, R. Motwani, and J. Widom, "Models and issues in data stream systems," in *Proceedings of the Twenty-first ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, ser. PODS '02. New York, NY, USA: ACM, 2002, pp. 1–16.

[10] "ConnectX®-3 Pro Single/Dual-Port Adapter with Virtual Protocol Interconnect," Mellanox, 2018. [Online]. Available: http://www.mellanox.com/page/products_dyn?product_family=161&mtag=connectx_3_pro_vpi_card

[11] "NUMA Balancing," Red Hat, 2018. [Online]. Available: https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/virtualization_tuning_and_optimization_guide/sect-virtualization_tuning_optimization_guide-numa-auto_numa_balancing