TPL: A Novel Analysis and Optimization Model for RDMA P2P Communication

Zhen Du, Zhongqi An, Jing Xing

TPL: A novel analysis and optimization model for RDMA P2P communication

Zhen Du[1,2], Zhongqi An[2], and Jing Xing[2]

[1] University of Chinese Academy of Sciences, China, Beijing
[2] High Performance Computer Research Center, Institute of Computing Technology, CAS, China, Beijing
duzhen18z@ict.ac.cn,{anzhongqi,xingjing}@ncic.ac.cn

Abstract. With increasing demand for networks with high throughput and low latency, RDMA is widely used because of its high performance. Because optimization for RDMA can fully exploit the performance potential of RDMA, methods for RDMA optimization is very important. Existing mainstream researches design optimization methods by constructing a more complete hardware view and exploring relation between software implementation and specific hardware behavior. However, the hardware architecture of NIC (like InfiniBand) is a "black box", which limits development of this type of optimization. So existing methods leave unsolvable problems. Besides, with development of RDMA technology, new features are proposed constantly. So, analysis and optimization methods of RDMA communication performance should be advancing with the times. The contributions of this paper are as follows: 1) We propose a new RDMA point-to-point communication performance analysis and optimization model: TPL. This model provides a more comprehensive perspective on RDMA optimization. 2) Guided by TPL, we design corresponding optimization algorithms for an existing problem, like WQE cache miss and a new scenario, like DCT. 3) We implement a new RDMA communication library, named ORCL, to put our optimizations together. ORCL eliminates WQE cache miss in real-time. And we simulate the workload of the in-memory KV system. Compared with existing RDMA communication implement, ORCL increases throughput by 95% and reduces latency by 10%.

Keywords: RDMA · Performance tuning · Optimization model.

1 Introduction

With the development of the in-memory key-value stores[8, 12], NVM distributed filesystem[11, 6], distributed deep learning systems[18, 7] and distributed graph processing system[14, 3, 20]. RDMA is widely used because of its high throughput and low latency. A well-optimized RDMA communication is related to low-level details, like hardware architecture and RDMA software options. And good optimization can increase performance by 10 times[9]. RDMA system designers need

to face many difficulties to improve RDMA performance. Besides, P2P communication is the bottom of the mainstream RDMA communication library[5, 1], which is the basis of RDMA optimization.

Many published results have analyzed and optimized RDMA P2P communication performance. They can be divided into two categories. One focus on traditional RDMA communication[9, 15, 19, 4], to exploit hardware view and find relation between RDMA options and low-level transmission process. The other is focus on new features of RDMA, like DCT[17, 16, 13]. These results follow the development of RDMA technology. They do tests and propose how to use these new features appropriately.

However, previous studies ignore following aspects: 1) The analysis and optimization of RDMA communication are not systematic enough. They only focus on hardware details, including CPU, NIC and PCIe, and transmission process of a single message. But they ignored dispatches of multiple messages and cooperation of different hardware. 2) RDMA new features like DCT(dynamically connected transport) and DM(device memory), should be optimized and included in RDMA communication optimization model. As far as we know, this article is the first one for DCT optimization and application of DM.

The contributions of this paper are as follows: 1) This paper presents a better analysis and optimization model of RDMA point-to-point communication——TPL. 2) Guided by TPL, we design algorithms to solve a remaining problem (WQE cache miss) and optimize a new feature (DCT). 3) We design and implement a new RDMA communication library——ORCL. In ORCL, WQE cache miss is eliminated. And we simulate workload of the in-memory KV system in our cluster. Compared with existing RDMA communication implement, ORCL increases throughput by 95% and reduces latency by 10%.

The rest of the paper is organized as follows: In Sect. 2, we provide the background information of RDMA. In Sect. 3, we propose TPL. In Sect. 4, we introduce two optimization cases guided by TPL. In Sect. 5, we evaluate our algorithm .

2 Background

2.1 Low-level Details Related to RDMA P2P Communication Performance

Fig. 1 shows hardware topology of RDMA P2P communication. PCIe root complex is core component of PCIe. It schedules communication between CPU, memory controller, and PCIe devices.

Fig. 2 shows the relation between main data structures and hardware. InfiniBand RDMA verbs is programming interface of RDMA communication. It contains several basic data structures: work queue element (WQE), queue pair (QP), completion queue (CQ), completion queue element (CQE)[2]. These data structures are stored in memory and some parts of them are cached in NIC cache[9].

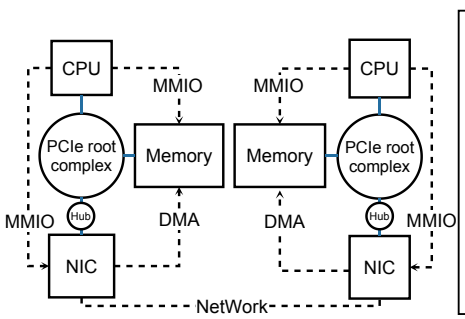

Fig. 1. Hardware Topology of P2P Communication

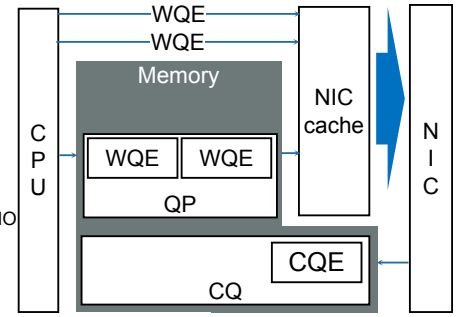

Fig. 2. Data Structures and Hardware

Verbs includes two main types of transport: RC (reliable connected transport), DCT (dynamically connected transport). RC promises the correctness and sequence of messages. Programmers need to establish a connection between QP before communication. DCT is a new protocol introduced by Mellanox in recent years. DCT has the same functionality as RC, but it can establish connections by hardware and connections can be changed. So DCT has much stronger scalability than RC.

2.2 Optimization Method Focusing on Hardware View and Single Message Transmission Process

Before reading this paper, we strongly recommend you to read this published guideline[9]. This guideline is basis of a series of RDMA P2P communication optimizations that have been proposed in recent years. It points out that there are a series of parameters and implementation details corresponding to different hardware behaviors that determine the performance of RDMA P2P communication, including several factors: 1) Transport flag, like *inlined* and *signaled*, 2) Transport type, like *RDMA write*, *RDMA read*, 3) Verbs type. This guideline constructs a more detailed hardware view and maps different communication implementations to different software and hardware processes for analysis and optimization.

3 TPL: Analysis and Optimization Model of RDMA P2P Communication Performance

TPL stems from some observations in our practice: 1) The opinions in existing guidelines have no way to explain all the phenomena we encounter. The factors that affect performance are more complicated. 2) The coordination between hardware also affects performance. 3) The key to solving the problem of performance degradation is sometimes irrelevant to specific transmission step that

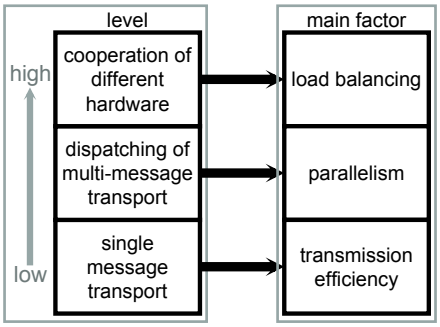

Fig. 3. Hardware Topology of P2P Communication

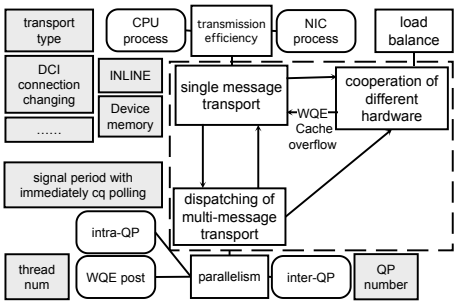

Fig. 4. Data Structures and Hardware

directly leads to degradation. 4) A new model is needed to face new features, like DCT.

As shown in Fig. 3, we found implementation communication will affect performance at three levels from low to high: single message transport, dispatching of multi-message transport, and coordination of different hardware. Optimization goals of different levels are different.

3.1 Transmission Efficiency of a Single Message

The transmission process of a single message includes steps of sending a message. Taking *RDMA write* as an example, the transmission process includes: 1) WQE is transferred from the CPU to the NIC, 2) payload is transferred from the memory to the NIC, 3) local NIC sends the message to remote NIC, 4) the payload is transferred from the remote NIC to remote memory.

3.2 Parallelism of WQE Submission and Handling

WQE submission and handling are respectively performed by CPU and NIC. Parallel processing can improve throughput of RDMA communication. WQE submission parallelism is determined by thread number of CPU.

There are two types of parallelism in WQE handling: intra-QP parallelism and inter-QP parallelism. The reason for intra-QP parallelism is the pipeline inside the process of WQE handling. Intra-QP parallelism is determined by signal period[9] (signaled WQE is submitted periodically, and CQ is polled to wait for the completion of the corresponding WQE). Inter-QP parallelism is determined by the number of QP and PU (QPs and PUs are bound, WQEs of different QPs can be handled by different PUs at the same time).

3.3 Load Balance between CPU and NIC

Throughput of RDMA communication meets bucket theory. Maintaining load balance between CPU and NIC can also improve performance. Because Infini-

Band NIC is a black box and only provides limited programming interfaces, the load balancing between the CPU and the NIC can only be adjusted indirectly.

3.4 Relation Between Three Dimensions

Only optimizing a certain dimension cannot maximize RDMA P2P communication performance. Adjustments to one dimension may affect other dimensions. So, three dimensions of TPL are not orthogonal and the relation between them is shown in Fig. 4.

4 Optimization Algorithm Design Based on TPL

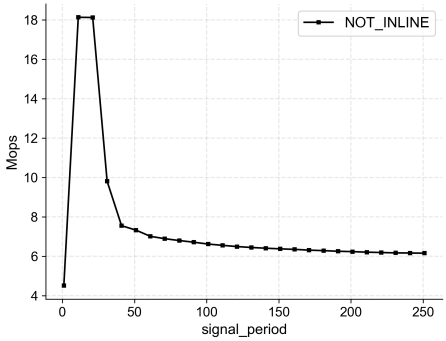

Fig. 5. WQE Cache Misses Recurrent. The Relation Between Signal Period and Throughput

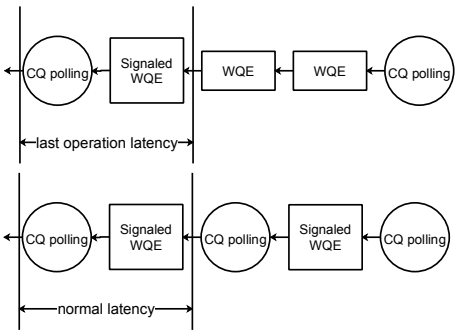

Fig. 6. Definition of Last Operation Latency and Normal Latency

TPL is the design principle and guideline of RDMA P2P communication performance optimization. This section will show two typical applications of TPL in the old problem (WQE cache miss) and a new scenario (DCT optimization).

4.1 TPL-based Systematic Analysis and Single-dimensional Optimization

WQE cache miss is a legacy problem that is difficult to solve in traditional verbs types. To increase RDMA communication performance, WQEs are cached in in-chip memory of NIC. When too many WQEs need to be cached, the WQE cache will be filled which causes WQE cache miss. WQE cache miss causes WQE re-fetching that lengthens transmission process and harm performance (as shown in Fig. 5). The existing method[9] focuses on WQE re-fetching detection which reduces performance directly. So existing researches use PCIe counter, but it is harmful to performance and not suitable to use in a production environment.

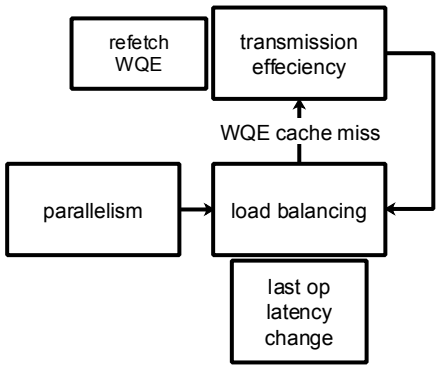

Fig. 7. Use TPL to Explain the Cause of WQE Cache Miss

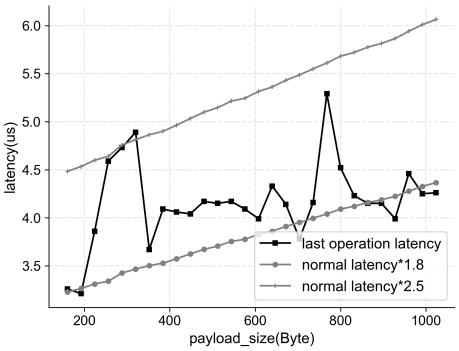

Fig. 8. The Relation between Last Operation Latency and Normal Latency with Best Parameters

Design of Dynamic Signal Algorithm based on TPL The design of dynamic signal algorithm is divided into two parts: WQE cache miss detection and WQE cache miss avoidance. Under the guidance of TPL, WQE cache overflow detection is easy. As shown in Fig. 7, increasing parallelism exacerbates the computing speed gap between fast CPU and slow NIC. Worse load balance between CPU and NIC makes many WQEs stay in cache, which causes WQE cache miss and reduces transmission efficiency. Lower transmission efficiency makes the situation of load balance even worse. Although the transmission process influences performance directly, load balance is the key to solve WQE cache miss. To analyze the situation of load balancing, we define two concepts (Fig. 6): 1) Last operation latency. When the signal period size is greater than 1, the interval between WQE submission at the end of the signal period and receiving the CQE corresponding to the WQE. Last operation latency is a characterization of load balance. 2) Normal latency. It is transport latency when signal period size is equal to 1. The normal latency is a reference used to determine whether last operation latency means WQE cache miss and bad load balancing.

As shown in Fig. 8, by adjusting the parameters, we can get good performance when the last operation latency is between 1.8*normal latency and 2.5*normal latency. WQE cache miss happens when last operation latency is greater than 2.5*normal latency. Last operation latency is less than 1.8*normal latency when NIC performance is not fully utilized.

WQE Cache Miss Avoidance According to TPL, developers can adjust the gap of running speed between CPU and NIC in multiple ways: number of threads, number of QPs, and signal period. Adjusting the signal period is the easiest. To improve the throughput as much as possible while preventing the overflow of the WQE cache, the signal period should be increased when the last operation latency is less than 1.8*normal latency. Correspondingly, when the final oper-

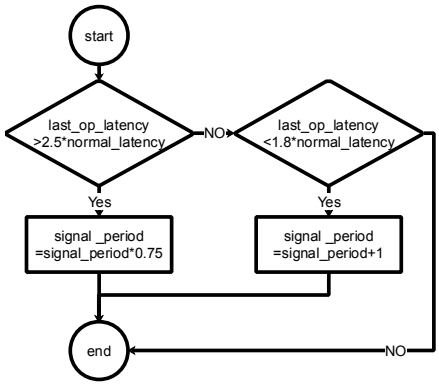

Fig. 9. The flow chart of dynamic signal period

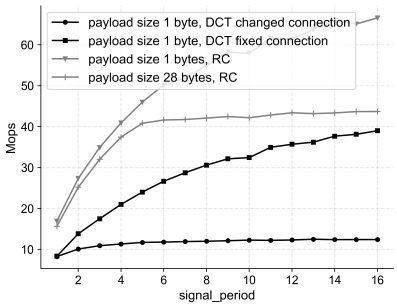

Fig. 10. Throughput comparison between DCT and RC

ating delay is higher than the 2.5*normal latency, the signal period should be reduced. WQE cache miss will affect other PCIe devices, the signal period should be conservatively increased and radically reduced. In this way, the last operation latency can be kept within a reasonable interval. The flow chart of this algorithm is shown in Fig. 9.

4.2 TPL-based Multi-dimensional Optimization

The technology of RDMA has been improving rapidly during the past decade. Besides, the HPC system based on RDMA is also highly customized. The designers of the RDMA system need to face new scenarios as and make optimization. This section will take new RDMA hardware and new verbs type (DCT) as an example to show how to optimize new scenarios under the guidance of TPL.

Testing and analysis of DCT and DM guided by TPL TPL simplifies the test design of RDMA communication. By separately fixing three of four factors, containing the transmission process, intra-QP parallelism, inter-QP parallelism as well as WQE-posting parallelism, the impact of these four factors on performance can be tested. And load balance also needs to be taken into consideration while analyzing test results.

Most test results of DCT are similar to RC. Compared with RC, the characteristics of DCT performance are mainly reflected in intra-QP parallelism. Figure 9 shows the throughput of RDMA write in different verbs types and signal period size. From the results of this test, the following conclusions can be revealed: 1) When the signal period is equal to 1, DCT throughput is less than RC, indicating that the DCT has higher latency and lower transmission efficiency. 2) Considering the size of the WQE header, the throughput of fixed connection DCT has a similar growth rate with RC. This comparison indicates that the two have similar intra-QP parallelism scalability, and lower transmission efficiency

is the only reason for the lower performance of fixed connection DCT. 3) The change of DCT connection does not affect latency, but it affects the growth rate of throughput, indicating that changing the connection will reduce the scalability of intra-QP parallelism. According to TPL, there are two optimization methods: 1) Directly relieving the decline of intra-QP parallelism. 2) Increasing inter-QP parallelism to make up for the decline in intra-QP parallelism.

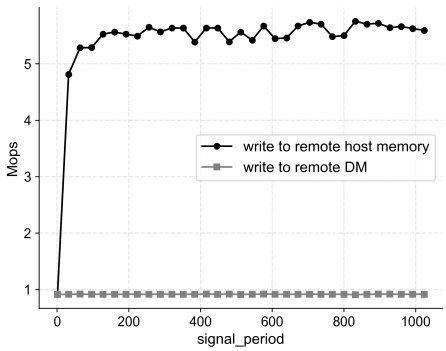

Fig. 11. Single Thread Read and Write Bandwidth to the Local Device Memory

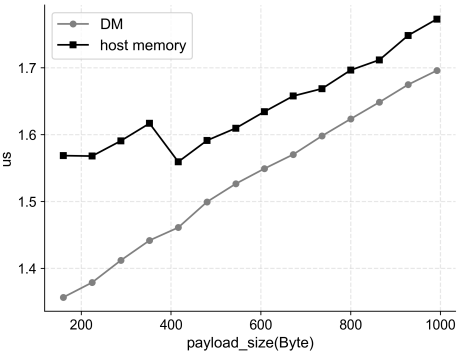

Fig. 12. Latency with Different Payload Size in Different Buffer Type

DM (device memory) is a new feature proposed in recent years. Users can explicitly copy the payload to the NIC by MMIO, which expands the way the payload is copied from memory to NIC. So, DM affects the transmission efficiency of a single message.

Fig. 13 shows the local device memory read and write performance. The local read and write performance is seriously asymmetric, and the DM write bandwidth is 200 times higher than the DM read bandwidth. This result shows that DM should not be used as a receive buffer to avoid reading DM.

Improving DCT single message transmission efficiency (T) When sending small messages, MMIO takes less time to copy the payload to NIC than DMA. Therefore, using DM as a sending buffer can shorten the transmission time of a single message. The test result shown in Fig. 12 shows that setting DM as send buffer reduces the latency by 15% while transmitting small messages.

Improving load balancing (L) According to the previous tests, changing DCT connection frequently causes the decrease of intra-QP parallelism scalability (will be mentioned below). Using DM as send buffer transfers part of the overhead from NIC to CPU (from DMA read to MMIO write), making the load between CPU and NIC more balanced when DCT connection is changed frequently. Figure 12 shows the effect of this optimization. When the DCT connection is frequently changed, DM can improve throughput by 30%.

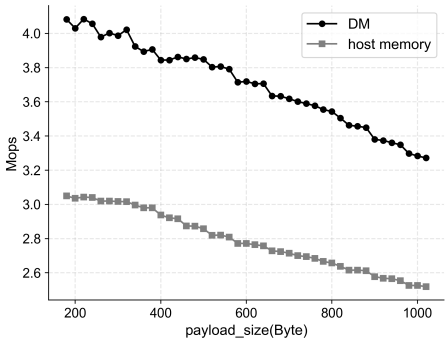

Fig. 13. Throughput of Different Payload Size and Send Buffer Type

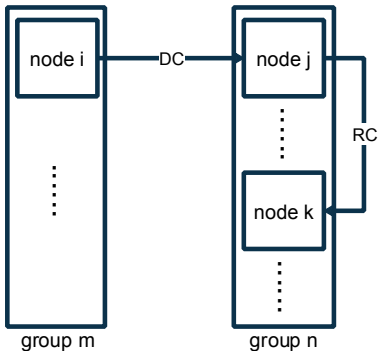

Fig. 14. Messages Sent From Node I to Node K

Improving the parallelism of DCT (P) We improve intra-QP parallelism. To reduce the probability of changing the DCT connection, all of the nodes are grouped. The nodes in the group are connected by RC. DCT is used for sending messages between groups. Sending messages to nodes outside the group requires forwarding, Fig. 14 shows this process. Node I from group m sends the message to node K from group N, and node J forwards this transmission. Node J and node I have the same index within their group.

5 Evaluation

ORCL is the RDMA communication library that we design under the guidance of TPL. ORCL-Classic is designed for mlx4 NIC, ORCL-Advanced is designed for mlx5 NIC. To prove that the optimization guided by TPL is effective, we test the effect of the dynamic signal period in ORCL-Classic on the elimination of WQE cache miss, and the performance of ORCL-Advanced delay under the workload of small-grain KV storage. The test platform of dynamic signal is two E5 nodes equipped with ConnectX-3, each node has 4 threads. The test platform of ORCL-Advanced is two E5 nodes equipped with ConnectX-5.

5.1 Testing and Analysis of Dynamic Signal Period

Avoidance of WQE cache miss is essentially an automatic tuning algorithm. Figure 9 shows the comparison of the throughput of dynamic signal period and fixed signal period under different fixed payload sizes. Fig. 15 shows that dynamic signal period has similar performance with best-parameters fixed signal period. Fig. 16 and Fig. 17 shows that the throughput of dynamic signal period is not lower than 90% of the best fixed signal period's throughput.

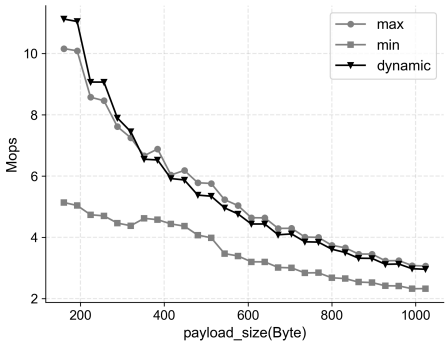

Fig. 15. The relation between payload and throughput of dynamic signal period, fixed signal period with best parameters as well as worst parameters

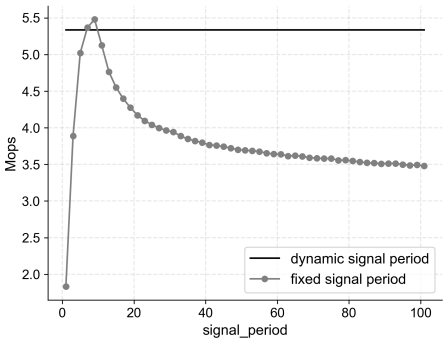

Fig. 16. The payload size is changed smoothly. Throughput of dynamic signal period and fixed signal period

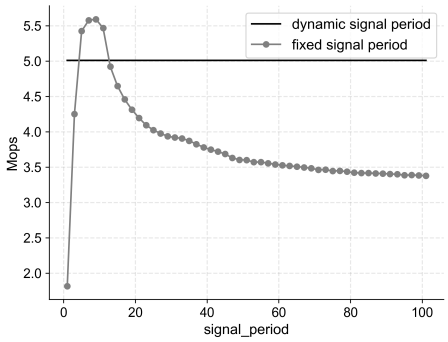

Fig. 17. Payload size is changed randomly. Throughput of dynamic signal period and fixed signal period

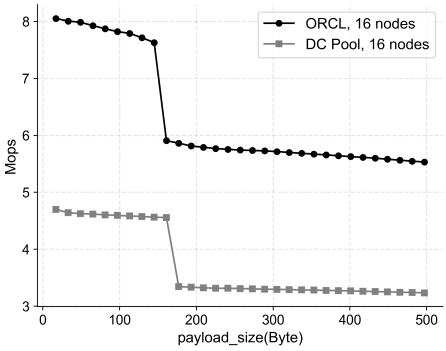

Fig. 18. Throughput of ORCL and traditional implement in the different payload

5.2 Throughput Test of ORCL-Advanced

Because the results of the latency test are similar to the result shown in figure 11, this subsection only shows the throughput of ORCL. We simulated a load of distributed hash-based in-memory KV storage with traces from YCSB. We simulated the scale of 16 nodes at two physical nodes. The test result is shown in Fig. 18. ORCL throughput is 95% higher than DC Pool (traditional implement designed by Hari Subramoni[16]).

6 Related Work

Designing high-performance RDMA systems is an active area of research. Anuj published three paper[8–10]. He revealed the effects of different verbs types,

transport types as well as designs on performance. And these researches propose a serious method to improve performance. ScaleRPC[4] improves the RDMA system scalability by improving CPU L3 cache hit rate. RFP[15] improves the performance when the number of clients is much higher than servers, by using different verbs in different transport direction between servers and clients.

Some works[17, 16, 13] for DCT test and research have been published. But these works don't include optimization of DCT. So, this paper is the first paper for DCT optimization.

7 Conclusion

In this article, we reveal the one-sidedness of existing RDMA communication optimization methods, and propose a new, more systematic analysis and optimization model: TPL. Under the guidance of this model, it is easier for us to find the key to solve the problem of performance degradation. The test results show that TPL can effectively solve the problem that existing methods are hard to solve. And TPL can has strong adaptability to new scenarios.

8 ACKNOWLEDGMENT

This work was supported by the National Key R&D Program of China (2018YFC0809300) and the National Natural Science Foundation of China (61502454).

And this work is supported in part by National Program on Key Research Project (No.2018YFB0204400), by NSFC (No.61702484,No.61972380), by CASSPRP (XDB24050200).

References

1. Openucx/ucx. https://github.com/openucx/ucx.
2. Dotan Barak. Verbs programming tutorial. *Open SHMEM*, 2014.
3. Rong Chen, Jiaxin Shi, Yanzhe Chen, Binyu Zang, Haibing Guan, and Haibo Chen. Powerlyra: Differentiated graph computation and partitioning on skewed graphs. *ACM Transactions on Parallel Computing (TOPC)*, 5(3):1–39, 2019.
4. Youmin Chen, Youyou Lu, and Jiwu Shu. Scalable rdma rpc on reliable connection with efficient resource sharing. In *Proceedings of the Fourteenth EuroSys Conference 2019*, pages 1–14, 2019.
5. Edgar Gabriel, Graham E Fagg, George Bosilca, Thara Angskun, Jack J Dongarra, Jeffrey M Squyres, Vishal Sahay, Prabhanjan Kambadur, Brian Barrett, Andrew Lumsdaine, et al. Open mpi: Goals, concept, and design of a next generation mpi implementation. In *European Parallel Virtual Machine/Message Passing Interface Users' Group Meeting*, pages 97–104. Springer, 2004.
6. Nusrat Sharmin Islam, Md Wasi-ur Rahman, Xiaoyi Lu, and Dhabaleswar K Panda. High performance design for hdfs with byte-addressability of nvm and rdma. In *Proceedings of the 2016 International Conference on Supercomputing*, pages 1–14, 2016.

7. Chengfan Jia, Junnan Liu, Xu Jin, Han Lin, Hong An, Wenting Han, Zheng Wu, and Mengxian Chi. Improving the performance of distributed tensorflow with rdma. *International Journal of Parallel Programming*, 46(4):674–685, 2018.
8. Anuj Kalia, Michael Kaminsky, and David G Andersen. Using rdma efficiently for key-value services. In *Proceedings of the 2014 ACM conference on SIGCOMM*, pages 295–306, 2014.
9. Anuj Kalia, Michael Kaminsky, and David G Andersen. Design guidelines for high performance {RDMA} systems. In *2016 {USENIX} Annual Technical Conference ({USENIX}{ATC} 16)*, pages 437–450, 2016.
10. Anuj Kalia, Michael Kaminsky, and David G Andersen. Fasst: Fast, scalable and simple distributed transactions with two-sided ({RDMA}) datagram rpcs. In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, pages 185–201, 2016.
11. Youyou Lu, Jiwu Shu, Youmin Chen, and Tao Li. Octopus: an rdma-enabled distributed persistent memory file system. In *2017 {USENIX} Annual Technical Conference ({USENIX}{ATC} 17)*, pages 773–785, 2017.
12. Christopher Mitchell, Yifeng Geng, and Jinyang Li. Using one-sided {RDMA} reads to build a fast, cpu-efficient key-value store. In *2013 {USENIX} Annual Technical Conference ({USENIX}{ATC} 13)*, pages 103–114, 2013.
13. Jiwoong Park, Yongseok Son, Heon Young Yeom, and Yoonhee Kim. Softdc: software-based dynamically connected transport. *Cluster Computing*, 23(1):347–357, 2020.
14. Jiaxin Shi, Youyang Yao, Rong Chen, Haibo Chen, and Feifei Li. Fast and concurrent {RDF} queries with rdma-based distributed graph exploration. In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, pages 317–332, 2016.
15. Maomeng Su, Mingxing Zhang, Kang Chen, Zhenyu Guo, and Yongwei Wu. Rfp: When rpc is faster than server-bypass with rdma. In *Proceedings of the Twelfth European Conference on Computer Systems*, pages 1–15, 2017.
16. Hari Subramoni, Khaled Hamidouche, Akshey Venkatesh, Sourav Chakraborty, and Dhabaleswar K Panda. Designing mpi library with dynamic connected transport (dct) of infiniband: early experiences. In *International Supercomputing Conference*, pages 278–295. Springer, 2014.
17. Masamichi Takagi, Norio Yamaguchi, Balazs Gerofi, Atsushi Hori, and Yutaka Ishikawa. Adaptive transport service selection for mpi with infiniband network. In *Proceedings of the 3rd Workshop on Exascale MPI*, pages 1–10, 2015.
18. Jilong Xue, Youshan Miao, Cheng Chen, Ming Wu, Lintao Zhang, and Lidong Zhou. Rpc considered harmful: Fast distributed deep learning on rdma. *arXiv preprint arXiv:1805.08430*, 2018.
19. Rohit Zambre, Megan Grodowitz, Aparna Chandramowlishwaran, and Pavel Shamis. Breaking band: A breakdown of high-performance communication. In *Proceedings of the 48th International Conference on Parallel Processing*, pages 1–10, 2019.
20. Xiaowei Zhu, Wenguang Chen, Weimin Zheng, and Xiaosong Ma. Gemini: A computation-centric distributed graph processing system. In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, pages 301–316, 2016.