Vrije Universiteit Amsterdam

Bachelor Thesis

RDMA Key Value store: Scalability of RDMA transportation types

---

Author: Jan Erik (Jens) Troost (2656054)

1st supervisor:	Animesh Trivedi
daily supervisor:	supervisor name
2nd reader:	supervisor name

A thesis submitted in fulfillment of the requirements for the VU Bachelor of Science degree in Computer Science

May 29, 2021

Abstract

Here goes the abstract of this thesis.

Contents

List of Figures

LIST OF FIGURES

List of Tables

LIST OF TABLES

1

Introduction

1.1 Context

Seemingly ever growing tech giants, such as Facebook, Amazon and Google, require fast, reliable and scalable key-value storage (KV-store) to serve product recommendations, user preferences, and advertisements. A KV-store is a data storage type which aims to store many small values and provides a high throughput and low latency interface. KV-stores usually offer a simple interface using 3 commands: GET, SET, and DEL. A trivial implementation makes use of a hash-table which matches the hash value of a given key, to the value associated with that key. Past advancements of KV-stores have focussed on improving these underlying data structures. For example, Cuckoo and Hopscotch hashing(2), have presented an alternative of an improved hash table algorithm. Nonetheless, with recent developments with RDMA networks, this has become a more attractive direction to improve KV-stores.

Remote direct memory access (RDMA) networks have increasingly become more popular in commercial and academic data centers, due to the increase in availability and decrease in cost for RDMA capable network interface cards (RNICs). RDMA offers a more direct connection between two machines, by interfacing the memory directly.

Instead of the operating system (OS) handling the incoming and outgoing packets, RNICs implement this in hardware, requiring no additional help from the OS and CPU. This results in lowering the overall latency and CPU overhead.

1.2 Problem statement

Making efficient use of RDMA networks is a difficult task, and requires in-depth knowledge on the hardware constraints present in the RNICs (1). Additionally, in a higher level sense, there are design choices to be made. RDMA networks can handle of various transportation types and so-called verbs. Each of these have advantage and disadvantages, in general setting, which can be read about in 2.3.

Furthermore, with the continuous growth of tech giants, an important factor is scalability. Some of these design choices impact scalability.

1.3 Research Question

This paper will explore to what extent RDMA transportation types affect the performance and scalability of KV-store. For this, this research will answer the following questions:

RQ1: What are the advantages and disadvantages of RDMA transportation types on an KV-store? This question aims to apply the already known information on the RDMA transportation types, on a RDMA KV-store specific setting. Aiding the design processes of further RDMA KV-store implementation, and possibly related implementations.

RQ2: How scalable are these transportation types?This being an important factor for tech giants which make use of RDMA KV-store, and require a system to be scalable and reach multiple clients.

1.4 Research Method

M1: A KV-store prototype will be implemented, with a flexible network interfacing, to include both socket and RDMA interfacing. This implementation can not only be used for this paper, but could also be expanded upon, for example with more advanced KV-store implementations.

M2: To investigate the performance and scalability between the various networking types, macro level benchmark will be designed, with realistic KV-store workloads.

A simple KV-store will be implemented. The backend, networking side, will be flexibly implemented to accommodate for the various transportation types.

Benchmarks will be done on DAS 5 computing cluster.

1.5 Plagiarism Declaration

1.6 Thesis Structure

Section 2 will provide the necessarily background knowledge of KV-store, linux sockets, and RDMA. Next, section 3 will go over the design of the KV-store, networking interfacing, and benchmark. In section 4 this design will be taken and described the implementation in a more technical prospective. The benchmarking result will be presented and analyzed in section 5. Section 6 compares findings with that of previously done work. Closing off the thesis, section 7 will go over the conclusion and provide recommendations.

2

Background

In this section, background information on key value stores, which is the backbone of this thesis. Also, the issues with commonly used linux sockets, will be explained. These issues have lead to the development of RDMA which is crucial for this thesis and research.

2.1 Key Value Store

2.2 Linux sockets

2.3 RDMA

Remote Direct Memory Access (RDMA) has been developed to address the issues of linux sockets. Providing a lower latency, less CPU overhead, and potentially higher throughput. The cost of RDMA is in the specialized hardware (RNICs) and added complexity.

As shown in section 2.2 above, the linux kernel has an extensive route, from NIC to application, including system calls and memory copies. RDMA achieves lower latency, and the reason why it requires specialized hardware, by bypassing the CPU and OS kernel when it comes to memory access (allowing for zero copy memory transfers) and packet handling.

2.3.1 Transportation types

Much like traditional networking, RDMA can make use of several transport types. Simply put, these can be split into two types, connection and datagram. Connection based can be split further into reliable and unreliable. Difference between reliable connection (RC) and unreliable connection (UC) is the use of ACK/NAK packets. With reliable transport

Transportation type	SEND	RECV	READ	WRITE	MTU size
RC	YES	YES	YES	YES	
UC	YES	YES	NO	YES	
UD	YES	YES	NO	NO	

Table 2.1: Verbs available to each transportation type

types these are sent, while with unreliable these are not. This could result in packet loss. However, it has been shown TODO: ADD REFERENCE, that this rarely occurs.

As stated in the name, RC and UC need a connection between queue pairs (QP). Only one QP can be connected to another QP. Contrasting this, unreliable datagram (UD) do not require a connection. This meaning that a UD QP can communicate with any other UD QP. UD therefore can make efficient use of a one-to-many network topology or application.

2.3.2 Verbs

To interact with RNICs, RDMA uses so-called "verbs" to execute specific types of instructions. Some of which are: read, write, send, and receive. Read and write follow so-called memory semantics, while send and receive follow channel semantics. Memory semantics require the destination memory address to be known. This meaning, to be able to perform a RDMA read of a remote memory location, the memory address of the requested memory needs to be known.

Channel semantics are simpler in the sense that the remote memory address does not need to be known. However, to perform a send operation from client to server, the server needs to first post a receive. This tells the server's RNIC which memory location the application expects the next incoming message to be place.

Not all verbs are available to every QP type. Table 2.1 summarizes the transportation type and which verb is available.

3

Design

...

4

Implementation

...

5

Evaluation

...

6

Related Work

...

7

Conclusion

...

7. CONCLUSION

References

[1] 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. 2

[2] Roxana Geambasu, Amit A Levy, Tadayoshi Kohno, Arvind Krishnamurthy, and Henry M Levy. Comet: An active distributed key-value store. In *OSDI*, pages 323–336, 2010. 1