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

2.1 Key Value Store

2.2 Linux sockets

2.3 RDMA

RDMA has been developed to address the issues of linux sockets. Providing a lower latency However, RDMA needs specialized hardware to

3

Designing a RDMA Key Value store

3.1 Key value store

This thesis is focused on the scalability using RDMA, and will not focus on advancing KV-stores. A simple KV-store is sufficient. There are two main and necessary commands needed: GET and SET. These commands will allow interaction with the hash table. The hash table internally uses a linked list.

3.1.1 Requests

For a client to interact with a KV server, the client has to send a request towards the server. A request is structured as shown in figure TODO MAKE FIGURE. Along with the command type, a key must be given. This key is used to identify the correct field within the hash table.

3.2 RDMA

For RDMA a design choice has to be made when designing a RDMA KV store, which verbs to use, and in turn which transportation type. With the focus of this thesis being the scalability of various transportation types, table TODO TABLE REF shows that using SEND and RECV would allow for all transportation types. A combination of transportation types could be used

3.3 Benchmark design

To evaluate the performance and scalability of RDMA KV store, there are a few concepts that need to be applied: multithreading and consistent. For scalability, it is important to

Cluster	Nodes	CPU type	CPU frequency (GHz	Memory (GB)	Network
VU	68	dual 8-core	2.4	64	IB and GbE
LU	24	dual 8-core	2.4	64	IB and GbE
UvA	18	dual 8-core	2.4	64	IB and GbE
TUD	48	dual 8-core	2.4	64	IB and GbE
UvA-MN	31	dual 8-core	2.4	64	IB and GbE
ASTRON	9	dual 8/10/14-core	2.6	128/512	IB, 40 GbE, GbE

Table 3.1: DAS-5 cluster specifications

evaluate the performance while increasing the number of clients. For this, multithreading is used to have multiple clients from a single node. This is used to

3.3.1 Baseline

As baseline, a TCP implementation will be used.

3.3.2 Experimental Setup

All performance tests and results have been gathered on the DAS-5 computing cluster. This distributed system of computers is spread across the Netherlands, and is used by research groups from VU Amsterdam, TU Delft, Leiden University, and many more. Each cluster varies slightly in specifications, however each, is equipped with a 48 Gbit/s Inifiniband (IB) RDMA networking, and 1 Gbit/s classical ethernet networking. The specifications of each cluster is as shown in table 3.1.

All experiments make use of the IB network card, and is also configured to run TCP/IP. Furthermore, at least two nodes are used. This ensures that the server and clients are separate.

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