An Empirical Evaluation of How The Network Impacts The Performance and Energy Efficiency in RAMCloud

Yacine Taleb, Shadi Ibrahim, Gabriel Antoniu, Toni Cortes

An Empirical Evaluation of How The Network Impacts The Performance and Energy Efficiency in RAMCloud

Yacine Taleb*, Shadi Ibrahim*, Gabriel Antoniu* and Toni Cortes†
**Inria Rennes Bretagne-Atlantique, Rennes, France*
†*Barcelona Supercomputing Center, Barcelona, Spain,*
†*Universitat Politècnica de Catalunya, Barcelona, Spain,*
*Email: *first.last@inria.fr, †first.last@bsc.es*

***Abstract*—In-memory storage systems emerged as a de-facto building block for today's large scale Web architectures and Big Data processing frameworks. Many research and engineering efforts have been dedicated to improve their performance and memory efficiency. More recently, such systems can leverage high-performance networks, e.g., Infiniband. To be able to leverage these systems it is essential to understand the trade-offs induced by the use of high-performance networks. This paper aims to provide empirical evidence of the impact of client's location on the performance and energy consumption of in-memory storage systems. Through a study carried on RAMCloud, we focus on two settings: 1) clients are collocated within the same network as the storage servers (with Infiniband interconnects); 2) clients access the servers from a remote network, through TCP/IP. We compare and discuss aspects related to scalability and power consumption for these two scenarios which correspond to different deployment models for applications making use of in-memory cloud storage systems.**

***Keywords*-In-memory storage, RAMCloud, Performance evaluation, Energy efficiency**

I. Introduction

Nowadays, web, mobile-apps, and gaming applications are being used by millions of users at the same time. Nevertheless, they must keep low latency access. To do so, service providers, such as Amazon [1], strongly rely on in-memory data stores and caches in order to keep low response times.

The increasing size of main memories has lead to the advent of new types of storage systems. These systems propose to keep all data in distributed main memories [2], [3], [4], [5]. For example, *Facebook* built a distributed in-memory key-value store using *Memcached* [6]. Twitter used *Redis* and scaled it to 105TB of RAM [7]. In addition to exploiting DRAM speed, they can offer: low-latency by relying on high speed networks [4], [2], [8]. For instance, RAMCloud leverages Infiniband and kernel-bypass to achieve durable writes in $15\mu s$. FaRM [9] utilizes RDMAs and thus can, in average, complete a *TCP-C* transaction in less than a millisecond.

High-performance networks are becoming more accessible in the Cloud [10] For example, Amazon is recently offering VM instances with the support of enhanced networking [11](i.e., by using the Amazon EC2 Elastic Network Adaptor with up to 20Gbps of aggregate network bandwidth). However, clients can access the storage in two setups: (1) in a HPC-like setup, clients access the storage system through the local network; (2) For Internet-based applications, which are typically hosted in the Cloud, clients access the system remotely through the typical TCP/IP stack. In this case clients will have slower access to the system, and more importantly it is not clear whether they will benefit or not from the speed of the in-memory storage.

We argue that the type of network used by clients to access the storage system impacts the performance. Moreover, given that main memories of DRAM-based servers consume between 25% up to 40% of the total energy of the servers [17], it is as vital to understand the network impact on energy efficiency as on performance.

With the increasing concerns about the energy issue in today's infrastructures [12], we propose in this work to study the impact of the network on performance and energy efficiency of a representative in-memory storage system, namely RAMCloud [2], [13]. RAMCloud is now used in various fields: analytics [14], or used as a low-latency storage for SDNs [15], or for scientific workflows [16]. Through an experimental study carried on GRID'5000 [17], we investigate the performance and energy consumption of RAMCloud storage system in both cases where clients are sharing the same network as the storage system and when they are not. We find that location of the clients could be a scalability and performance limiting factor of in-memory storage systems. For example, 10 RAMCloud servers obtain a throughput of up to 2 Million op/s when accessed by 90 clients who are using Infinband, while the throughput is limited to 200K op/s for the same settings but when the clients are connected to the cluster using TCP/IP.

The remainder of this paper is organized as follows: Section II discusses why we chose RAMCloud and describes it. Section III presents our experimental configuration and the main findings of our experiments. Section V presents related work. Finally, Section VI concludes this work.

II. Background

In this section we motivate the choice of RAMCloud and present its main features. Finally, we present a motivating example on the relevance of our study.

A. A representative system: RAMCloud

Ideally, the main attributes that in-memory storage systems should provide are performance, durability, availability, scalability, efficient memory usage, and energy efficiency. Most of today's systems target performance and memory efficiency [18], [19], [20]. Durability and availability are also important as they avoid application developers to backup in-memory data in secondary storage and handle the synchronization between the two levels of storage. On the other hand, scalability is vital, especially with today's high-demand Web applications. They are accessed by millions of clients in parallel. Therefore, large scale clustered storage became a natural choice to cope with such high demand[6].

In contrast with most of the recent in-memory storage systems, RAMCloud main claims are performance (low-latency), durability, scalability, and memory efficiency. The other closest system to provide all these features to be found in the literature is FaRM [4]. Unfortunately it is neither open-source nor publicly available.

B. The RAMCloud storage system

RAMCloud is an in-memory key-value store. The main claims of RAMCloud are low-latency, fast-crash recovery, and efficient memory usage. Operations on small objects can be achieved in a few microseconds by using high performance networks (e.g., Infiniband). Fast-crash recovery is achieved by randomly replicating data to as many backups as possible in the cluster then reconstructing data by exploiting in parallel backups and recovery servers. Memory efficiency is achieved through using DRAM as a log-structured memory and through efficient memory cleaning. Next we present the main concepts and mechanisms of RAMCloud.

Architecture A RAMCloud's cluster consists of three entities : a coordinator maintaining meta-data information about storage servers, backup servers, and data location; a set of storage servers that expose their DRAM for the clients as storage space; and backups that will store replicated data in their DRAM temporarily and persist it to disk asynchronously. Usually, storage servers and backups are collocated. Thus a physical machine will run both storage and backup services at the same time.

Data structures Data in RAMCloud is stored in a set of tables. Each table can span multiple storage servers. A table is partitioned into a set of *tablets*. A uniform hash function distributes evenly the objects across tablets.

A server in RAMCloud uses a log-structured memory to store its data. It uses a hash-table to index data. This log-structured approach, coupled with a two level cleaning approach, enable RAMCloud to use DRAM efficiently.The log-structured memory of each server is divided into 8MB segments. A server stores data in an append-only fashion. Thus, to free unused space a log-cleaner is called whenever a server reaches a certain memory utilization threshold. The cleaner copies a segment's live data into the free space (still available in DRAM) and removes the old segment. As the cleaner is called more often in memory, when data on disk grows larger than a threshold the two-level cleaning starts, this time cleaning data in both memory and disk at the same time.

Data durability: It is one of the most important aspects of RAMCloud, as it has impacted most of its design. In RAMCloud data is present all time in DRAM. However, durability is guaranteed by replicating data to remote disks. More precisely, whenever a storage server receives a write request, it appends the object into its latest free segment, and forwards a replication request to the backup servers randomly chosen for that segment. The server waits all acknowledgements from backup servers to answer client's update request. Backup servers will keep a copy of this segment in DRAM until it fills, then flush the segment to disk and remove it from DRAM. For each new segment, a random backup in the cluster is chosen. This enables RAMCloud to harness large-scale to enable fast crash recovery. Moreover, when a server crashes, multiples servers will be involved in the crash recovery, each one of them taking responsibility of a set of segments to ensure even re-distribution of the recovered data.

C. Location of the clients: Why is it important?

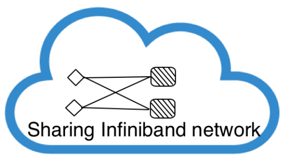
To understand the importance of the network, we plot Figure 1 that shows the two possible ways of using RAMCloud. In Figure 1a the clients are sharing the same datacenter network as the storage system. In this case the clients can take full advantage of the storage system and benefit from high speed networks (when available). A typical example is a HPC setup or in-memory BigData analytics [21]. On the other hand, a second use case is displayed in Figure 1b, where the clients use typical a TCP/IP network stack to access the storage system. In this case clients will have slower access to the system, and more importantly it is not clear whether they will benefit or not from the speed of the in-memory storage. This is the case for a Cloud deployment for instance.

Through an experimental study, we attempt to answer this question by reproducing both scenarios displayed in Figure 1. In Section IV we present insights about the performance and energy efficiency of RAMCloud in both cases.

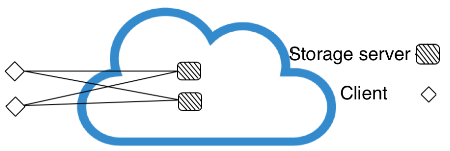
III. Experimental evaluation

A. Benchmark

We used the industry standard Yahoo! Cloud Serving Benchmark (YCSB) benchmarking framework [22]. YCSB is an open and extensible framework that allows to mimic



(a) Clients and servers share the same network



(b) Clients access servers from outside the datacenter

Figure 1: The two possible scenarios for accessing RAMCloud. Figure 1a refers to the case where the clients also use Infiniband transport. Figure 1b represents the case where the clients are not collocated within the same datacenter with the storage servers. We refer to this case in RAMCloud as the one where the client uses TCP/IP

real-world workloads such as large scale web applications, to benchmark key-value store systems. YCSB supports a large number of databases and key-value stores, which is convenient for comparing results with different systems.

Basically, one needs to fill the data-store with a YCSB client. It is possible to specify the request distribution, i.e., uniform, zipfian, etc. Executing the workload consists of running clients with a given workload specification, i.e., number of requests, distribution of requests, number of threads per clients, etc.

B. *Platform*

The experiments were performed on the Grid'5000 [17] testbed. The Grid'5000 platform provides researchers with an infrastructure for large-scale experiments. It includes 9 geographical sites spread across French territory and 1 located in Luxembourg. We relied on Nancy's site nodes to carry our experiments. More specifically, the nodes we used have 1 CPU Intel Xeon X3440, 4 cores/CPU, 16GB RAM, and 298GB HDD. Additionally each one has an Infiniband-20G and a Gigabit Ethernet card.

We have chosen these nodes as they offer capability to monitor power consumption: 40 of these nodes are equipped with PDUs which allow to retrieve power consumption through an SNMP request.

Throughout all our experiments we make sure to reserve the whole cluster which consists of 133 nodes, to avoid any interference with other users of the platform. We dedicate the 40 nodes equipped with PDUs to run RAMCloud's cluster, i.e., master and backup services. We run YCSB clients on the 90 different nodes to avoid any interference of any sort. A single node is used to run the coordinator service.

C. *Experiments' Configuration*

In our experiments each client runs on a single machine, this helps us avoid interferences of any sort. Each node ran as server and backup at the same time. We have fixed the memory used by a RAMCloud server to 10GB and the available disk space to 80GB. We have fixed the memory size to an acceptable limit to carry the whole data in the cluster. Before running each benchmark, we pre-load 100K records of 1KB in the cluster. Running the benchmark consists of launching simultaneously one instance of a YCSB client on each client node. Each client issues 100K requests, which corresponds to the total number of records. Having an increasing number of requests with the number of clients enabled us to study concurrency impact's on RAMCloud's scalability. Having each client generate 100K requests results in having 1M requests with 10 clients for example, and 9M requests with 90 clients, which corresponds to 8.58GB of data requested per run.

It is noteworthy that we use read-only workloads in our experiments. Since our focus is on understanding how the client location can impact the energy efficiency of the system, read-only workloads are more suitable. Figure 2 shows the sequence diagram of update and read operations in RAMCloud. When a client issues an update request it is first processed by the server holding primary-replica. Then the server creates a replication request to the servers holding the secondary-replicas of the object being written. The client gets the acknowledgement only after all secondary-replicas have replied to the primary-replica. Therefore, update/insert operations are not suitable in our study because of the intra-cluster communication they induce between primary- and secondary-replicas. On the other hand, read operations are processed only by the primary-replica. In this case, there is a client-server communication exclusively, which is better suited for our study.

In our figures, each value corresponds to an average of 5 runs with the corresponding error bars. When changing the cluster configuration (i.e., number of RAMCloud servers), we remove all data stored on servers as well as in backups, then we restart the RAMCloud servers and backups on the whole cluster to avoid any interference with prior experiments.

We configured RAMCloud with the option *ServerSpan* equal to the size of the cluster. As RAMCloud does not have a smart data distribution strategy, this option is used to manually decide the distribution of data across the servers, i.e., how many servers each table will span. Moreover, we used uniform data distribution in YCSB clients to insert and request data. At the end, all the data is evenly distributed across the servers and each object is requested at the same frequency.

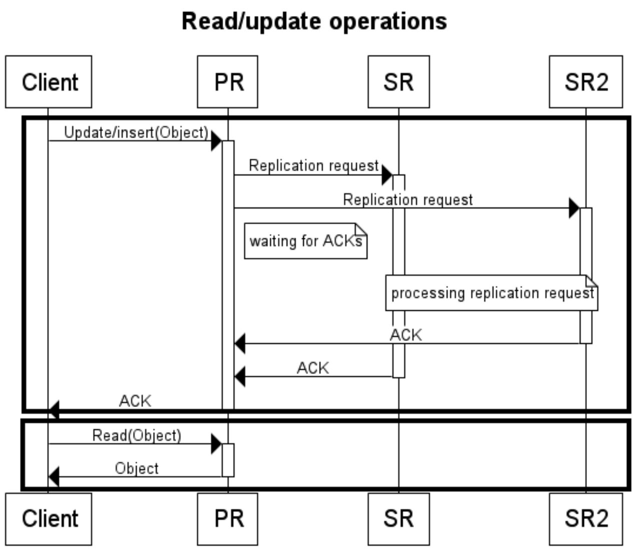


Figure 2: Sequence diagram of read and update/insert operations in RAMCloud. PR and SR stand for primary-replica and secondary-replica respectively. Every entity is a running on a distinct machine in the cluster.

IV. Results

A. Performance and scalability

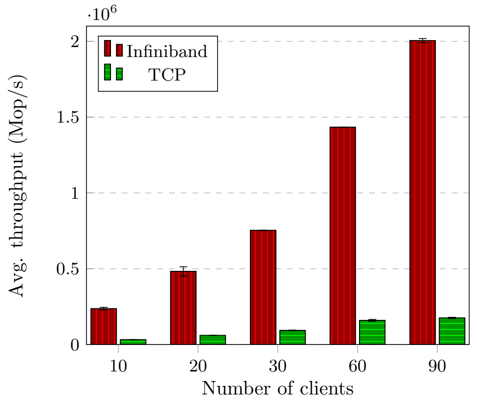


Figure 3: Total aggregated throughput as a factor of clients number with a fixed cluster size of 10 nodes.

Peak performance: Figure 3 shows the aggregated throughput for a 10 node cluster with both Infiniband and TCP. A considerable gap can be observed in the throughput achieved when clients access the system with Infiniband and TCP. When running 10 clients with Infiniband, the system achieves 7.38x more throughput compared to when clients use TCP. Whenever increasing the number of clients the gap widens and reaches 11.40x more throughput for Infiniband cluster. What stands out here is the limited scalability of TCP, for example between 60 and 90 clients while Infiniband increases by a factor of 1.39x TCP only increases by 1.10x from 60 to 90 clients. This suggests that since the transport protocol is slowing down the operations, RAMCloud nodes are subject to more concurrency, and thus scalability might suffer.

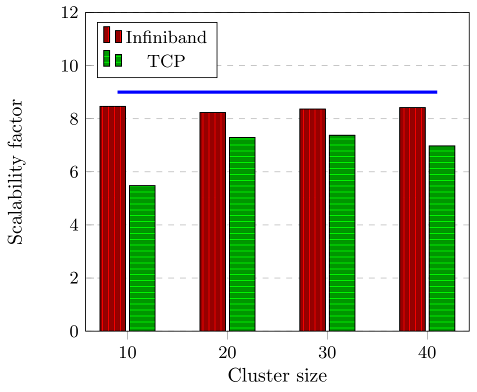


Figure 4: Throughput scalability factor as a function of cluster's size and when running 90 clients. The blue line represents the expected increase in throughput according to the baseline of 10 clients.

Scalability: To further investigate that behaviour we plot in Figure 4 the scalability of each of the scenarios, i.e., varying the cluster size from 10 to 40 nodes and we fix the clients to 90. The figure represents the ratio of throughput increase when taking 10 clients as a baseline. When clients are collocated with RAMCloud, it can achieve linear scalability as we already showed in the previous section, even under high concurrent accesses. On the other hand, we can witness the difference widening between TCP scalability and the expected ratio. For the cluster of 10 nodes with TCP the scalability factor is 5.4 while the expected scalability factor is 9. This difference reduced when increasing the cluster size, for example it is 6.9 for 20 nodes and 7.3 for 30 nodes. This confirms our observation about RAMCloud being less scalable in throughput when accessed by TCP compared to Infiniband. We suspect this is due to lesser load on resources as we increase cluster size.

As expected, when clients are collocated on the same network as RAMCloud it can achieve more throughput compared to when clients access the system from a different network. In the same lines, when clients are collocated with RAMCloud it achieves better scalability.

B. Per-node power consumption

Energy overhead of RAMCloud: Figure 5 represents the average power consumption of the cluster for the same experiment described above. First, it is important to notice the horizontal blue bar that represents the average power consumed by the machines when idle, which is in average 50 Watts. The horizontal black dashed line represents the average power per server when running the RAMCloud service. There is an increase of 1.6X in the average power consumed per node since RAMCloud hogs one core per node, even when idle. This is due to the *polling mechanism*

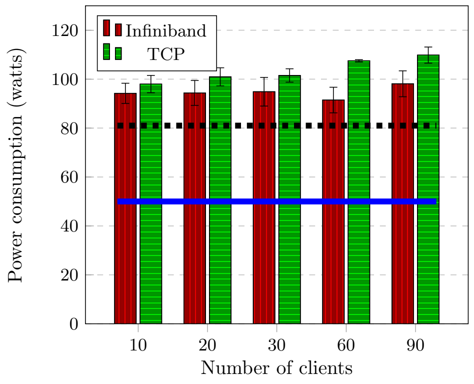


Figure 5: Average power consumption per server in Watts. The cluster size is fixed to 10. The horizontal blue line represents the average power consumed by servers when idle. The horizontal dashed black line represents the average power per server when RAMCloud is running and idle.

Clients \ Transport	TCP	IB
	min — max	min — max
0	25 — 25	25 — 25
10	**52,933** — 54,543	44,193 — 53,391
20	**55,044** — 57,244	46,985 — 57,921
30	**56,638** — 58,943	40,534 — 46,960
60	**59,917** — 62,234	47,974 — 62,235
90	**65,129** — 67,676	50,626 — 65,958

Table I: The minimum and maximum of the average CPU usages (in percentage) of all servers when running read-only workload on 10 RAMCloud servers.

of RAMCloud, which polls continuously requests from the NIC in order to handle packets as fast as possible.

Impact of clients' location on the power consumption: In figure 5, when clients use Infiniband, we can see a stable power consumption of 94Watts up to 60 clients, while it increases a little bit when going up to 90 clients. More likely, this is the point where the cluster starts demanding more resources to cope with the load, which matches our observation from section 1. If we look at the TCP cluster power consumption the results are more surprising. First we see that the average power consumption is increasing constantly from 97Watts for 10 clients up to 109Watts for 90 clients. Moreover, in all cases it is higher than the Infiniband cluster average power consumption.

More CPU usage when using TCP: To understand why the system consumes more power when accessed by TCP compared Infiniband, we show table I. It represents the minimum and maximum CPU usage with different numbers of clients for a cluster of 10 RAMCloud servers. We remark a slight additional maximum CPU usage when clients use TCP compared to IB. What is more interesting is to see the minimum CPU usage of the servers when clients use TCP (bold) compared to when they use IB. There is between 9% additional CPU usage and up to 15%. We have noted that in some scenarios, servers reach the same maximum CPU usage. More importantly, we observed high variation in CPU usage in the case of Infiniband. For example, when running 10 clients, there is a 9% difference between the minimum and maximum CPU usage when clients are using Infiniband.

As a result, when clients use TCP to access RAMCloud, the system exhibits higher CPU usage compared to when accessed through Infiniband. This increased CPU usage results in additional average per-node power consumption.

C. The energy efficiency

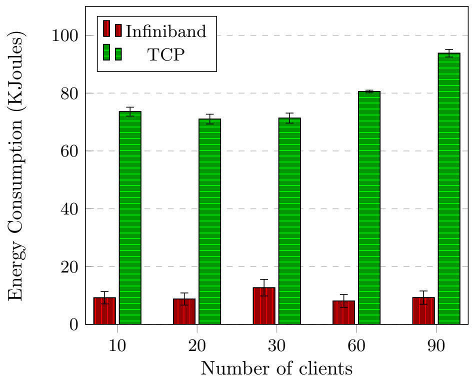


Figure 6: The total energy consumption when running read-only workload with a cluster of 10 nodes both with TCP and Infiniband

Impact on overall energy consumption: We plot figure 6 to show the overall energy consumption when clients use both Infiniband and TCP. Up to 60 clients the average difference is roughly 7X more energy consumed when clients use TCP. The difference goes up to 9X when running 90 clients.

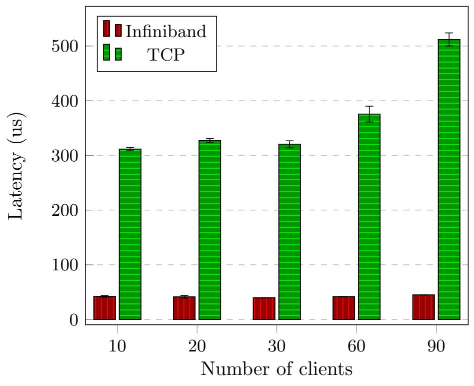


Figure 7: The latency per single operation when running read-only workload with a cluster of 10 nodes both with TCP and Infiniband

Latency per operation: While figure 5 has shown that accessing the system through TCP can result in more average per-node power consumption compared to when accessed through Infiniband, the expected difference in total energy consumption is not as big as shown in figure 6. To explain such a phenomena we plot Figure 7 which represents the latency of a single operation for the same scenario, i.e., 10 servers running up to 90 clients. When looking into Infiniband's cluster latency one can see a stable latency around 40 μs. For TCP cluster the latency is stable around 315 μs up to 60 clients. It reaches more than 512 μs for 90 clients.

Consequently, when a client issues a request with TCP it takes longer to be processed at the server level compared to when it is issued with Infiniband. While this is not surprising, it can explain the wide gap that appears in figure 6 between the cases where clients access RAMCloud through TCP and Infiniband.

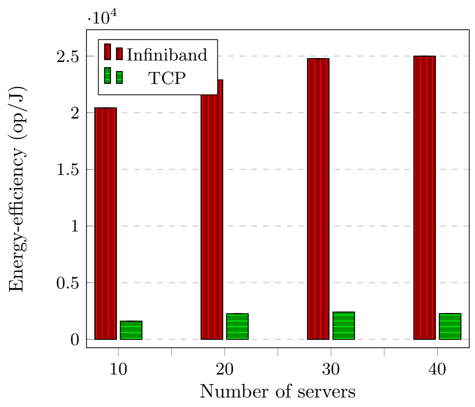


Figure 8: The energy efficiency when running read-only workload with different number of servers both with TCP and Infiniband. The number of clients is fixed to 90.

The energy efficiency Figure 8 represents the energy efficiency of different number of RAMCloud servers when running 90 clients. Consequently to the increased power consumption and high latency when clients access the system with TCP, the energy efficiency is very low compared to when clients use Infiniband. In average the system has 10X better energy efficiency when accessed with Infiniband compared to TCP.

To summarize, we find that RAMCloud has a better energy efficiency when clients use Infiniband network to access it compared to when they use TCP/IP. The reasons are that when clients access RAMCloud through TCP, the system has a higher CPU usage compared to when accessed through Infiniband. Moreover, using TCP leads to higher latencies, therefore the system has a much higher energy consumption compared to when using Infiniband. All these reasons contribute to the energy-inefficiency of the TCP transport in RAMCloud.

V. Related work

Many research efforts have been dedicated to improve the performance of in-memory storage systems. However, very little visibility exists on their energy efficiency. More specifically, the relation between the type of the network used to access the system and the energy efficiency has not been explored. Yet, there have been a lot of works on improving the energy efficiency of disk-based storage systems.

In-memory storage systems More and more companies are adopting in-memory storage systems. *Facebook* leveraged Memcached to build a distributed in-memory key-value store. Twitter used *Redis* and scaled it to 105TB of RAM. Alternatively, a lot a academic work has been proposed pushing the limits of performance of storage systems to a next level. As an example *MemC3* [19] is an enhancement of Memcached. Through a set of engineering and algorithmic improvements, it outperformed the original Memcached by 3x in terms of throughput. *Mica* [20] is an in-memory key-value store that takes advantage of RDMAs (Remote Direct Memory Access) and better parallelism handling. Similarly, *FaRM* [4] is a distributed in-memory storage system based on RDMAs and proposes transactional support.

Evaluating performance and energy efficiency of storage systems Many studies have been conducted to characterize the performance of storage systems and their workloads. In [23] a deep analysis is made on traces from Facebook's Memcached deployment and gives insight about the characteristics of a large scale caching workload. In another dimension, the work in [24] proposes to study the throughput of six storage systems. In [25] authors explore the energy-consistency trade-off and reveal that energy can vary considerably according to the level of consistency.

However, our work complements the previous studies as it shows that the location of the clients can play an important role in the performance and energy efficiency of in-memory storage systems.

VI. Conclusion and future work

In this paper we presented our efforts in understanding the main factors impacting performance and energy efficiency in in-memory storage system. We zoomed into the networking technology impact on the RAMCloud storage system.

We confirmed foreseen results such as the scalability of RAMCloud's throughput when using Infiniband. However, we discovered that using RAMCloud with TCP does not scale as well as with Infiniband. This phenomena becomes dramatic at high loads. Moreover, we discovered that using TCP leads to more energy consumption compared to using Infiniband. When investigating the issue, we remarked that it was due to higher CPU occupation times whenever running with TCP.

As future work we plan to complete the picture of the main factors impacting performance and energy efficiency. To do so we plan to break down the design principles of RAMCloud such as the polling mechanism, log-structured memory, replication, crash-recovery mechanism, and study the impact of each of them on the performance and energy efficiency. Doing so can help system designer to build more energy-aware systems in the future. It can also open new research opportunities in investigating the trade-offs between energy efficiency and performance in in-memory storage systems.

Our ultimate goal would be to model the performance and energy consumption for in-memory storage systems. While this is a very hard task, it can have a huge impact as it can enable system designers to be aware of the impact of each feature in their system on the performance and energy consumption. Moreover, it can help system administrators tune their system in order to achieve better energy efficiency.

Acknowledgment

This work has been supported by the BigStorage project, funded by the European Union under the Marie Sklodowska-Curie Actions (H2020-MSCA-ITN-2014-642963), by the Spanish Government (grant SEV2015-1305 0493 of the Severo Ochoa Program), by the Spanish Ministry of Science and Innovation (contract TIN2015-65316), and by Generalitat de Catalunya (contract 2014-SGR-1051).

References

[1] "Amazon ElastiCache," https://aws.amazon.com/elasticache/, 2017, [Online; accessed January-2017].

[2] D. Ongaro, S. M. Rumble, R. Stutsman, J. Ousterhout, and M. Rosenblum, "Fast crash recovery in ramcloud," in *Proceedings of the Twenty-Third ACM Symposium on Operating Systems Principles*, ser. SOSP '11, 2011, pp. 29–41.

[3] "Riak," www.basho.com/products/riak-kv/, 2017, [Online; accessed January-2017].

[4] A. Dragojević, D. Narayanan, M. Castro, and O. Hodson, "Farm: Fast remote memory," in *11th USENIX Symposium on Networked Systems Design and Implementation (NSDI 14)*, Seattle, WA, Apr. 2014, pp. 401–414.

[5] "VoltDB," https://www.voltdb.com/, 2017, [Online; accessed January-2017].

[6] R. Nishtala, H. Fugal, S. Grimm, M. Kwiatkowski, H. Lee, H. C. Li, R. McElroy, M. Paleczny, D. Peek, P. Saab, D. Stafford, T. Tung, and V. Venkataramani, "Scaling memcache at facebook," in *Presented as part of the 10th USENIX Symposium on Networked Systems Design and Implementation (NSDI 13)*, Lombard, IL, 2013, pp. 385–398.

[7] M. Iravani, "How twitter uses redis to scale - 105tb ram, 39mm qps, 10,000+ instancess," 2015. [Online]. Available: https://www.linkedin.com/pulse/how-twitter-uses-redis-scale-105tb-ram-39mm-qps-10000-iravani

[8] Y. Wang, L. Zhang, J. Tan, M. Li, Y. Gao, X. Guerin, X. Meng, and S. Meng, "Hydradb: A resilient rdma-driven key-value middleware for in-memory cluster computing," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, ser. SC '15, 2015, pp. 22:1–22:11. [Online]. Available: http://doi.acm.org/10.1145/2807591.2807614

[9] Aleksandar Dragojevic, Dushyanth Narayanan, "No compromises: distributed transactions with consistency, availability, and performance," in *Symposium on Operating Systems Principles (SOSP'15)*, 2015.

[10] "Microsoft Azure Cloud," https://azure.microsoft.com/pricing/details/cloud-services/, 2017, [Online; accessed January-2017].

[11] "Amazon enhanced networking," https://aws.amazon.com/ec2/instance-types/enhanced$_networking$, 2017, $[Online; accessedJanuary-$ 2017].

[12] "Energy consumption in US Datacenters," http://www.nrdc.org/energy/data-center-efficiency-assessment.asp, 2016, [Online; accessed January-2017].

[13] S. M. Rumble, A. Kejriwal, and J. Ousterhout, "Log-structured memory for dram-based storage," in *Proceedings of the 12th USENIX Conference on File and Storage Technologies (FAST 14)*, Santa Clara, CA, 2014, pp. 1–16.

[14] C. Tinnefeld, D. Kossmann, M. Grund, J.-H. Boese, F. Renkes, V. Sikka, and H. Plattner, "Elastic online analytical processing on ramcloud," in *Proceedings of the 16th International Conference on Extending Database Technology*, ser. EDBT '13, 2013, pp. 454–464.

[15] P. Berde, M. Gerola, J. Hart, Y. Higuchi, M. Kobayashi, T. Koide, B. Lantz, B. O'Connor, P. Radoslavov, W. Snow, and G. Parulkar, "Onos: Towards an open, distributed sdn os," in *Proceedings of the Third Workshop on Hot Topics in Software Defined Networking*, ser. HotSDN '14, 2014, pp. 1–6.

[16] J. Blomer and G. Ganis, "Large-scale merging of histograms using distributed in-memory computing," *Journal of Physics: Conference Series*, vol. 664, no. 9, p. 092003, 2015.

[17] "GRID'5000," www.grid5000.fr/, 2017, [Online; accessed January-2017].

[18] C. Mitchell, Y. Geng, and J. Li, "Using one-sided rdma reads to build a fast, cpu-efficient key-value store," in *Presented as part of the 2013 USENIX Annual Technical Conference (USENIX ATC 13)*, San Jose, CA, 2013, pp. 103–114.

[19] B. Fan, D. G. Andersen, and M. Kaminsky, "Memc3: Compact and concurrent memcache with dumber caching and smarter hashing," in *Presented as part of the 10th USENIX Symposium on Networked Systems Design and Implementation (NSDI 13)*, Lombard, IL, 2013, pp. 371–384.

[20] H. Lim, D. Han, D. G. Andersen, and M. Kaminsky, "Mica: A holistic approach to fast in-memory key-value storage," in *11th USENIX Symposium on Networked Systems Design and Implementation (NSDI 14)*, Seattle, WA, Apr. 2014, pp. 429–444.

[21] "Apache Spark," https:// http://spark.apache.org/, 2017, [Online; accessed January-2017].

[22] B. F. Cooper, A. Silberstein, E. Tam, R. Ramakrishnan, and R. Sears, "Benchmarking cloud serving systems with ycsb," in *Proceedings of the 1st ACM Symposium on Cloud Computing*, ser. SoCC '10, 2010, pp. 143–154.

[23] B. Atikoglu, Y. Xu, E. Frachtenberg, S. Jiang, and M. Paleczny, “Workload analysis of a large-scale key-value store,” in *Proceedings of the 12th ACM SIGMETRICS/PERFORMANCE Joint International Conference on Measurement and Modeling of Computer Systems*, ser. SIGMETRICS ’12, 2012, pp. 53–64.

[24] T. Rabl, S. Gómez-Villamor, M. Sadoghi, V. Muntés-Mulero, H.-A. Jacobsen, and S. Mankovskii, “Solving big data challenges for enterprise application performance management,” *Proc. VLDB Endow.*, vol. 5, no. 12, pp. 1724–1735, Aug. 2012.

[25] H. E. Chihoub, S. Ibrahim, Y. Li, G. Antoniu, M. S. Prez, and L. Boug, “Exploring energy-consistency trade-offs in cassandra cloud storage system,” in *2015 27th International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD)*, 2015, pp. 146–153.