About Microservices, Containers and their Underestimated Impact on Network Performance

Nane Kratzke

Lübeck University of Applied Sciences, Center of Excellence CoSA
Lübeck, Germany
email: nane.kratzke@fh-luebeck.de

***Abstract*—Microservices are used to build complex applications composed of small, independent and highly decoupled processes. Recently, microservices are often mentioned in one breath with container technologies like Docker. That is why operating system virtualization experiences a renaissance in cloud computing. These approaches shall provide horizontally scalable, easily deployable systems and a high-performance alternative to hypervisors. Nevertheless, performance impacts of containers on top of hypervisors are hardly investigated. Furthermore, microservice frameworks often come along with software defined networks. This contribution presents benchmark results to quantify the impacts of container, software defined networking and encryption on network performance. Even containers, although postulated to be lightweight, show a noteworthy impact to network performance. These impacts can be minimized on several system layers. Some design recommendations for cloud deployed systems following the microservice architecture pattern are derived.**

Keywords–Microservice; Container; Docker; Software Defined Network; Performance

I. INTRODUCTION

Microservices are applied by companies like Amazon, Netflix, or SoundCloud [1] [2]. This architecture pattern is used to build big, complex and horizontally scalable applications composed of small, independent and highly decoupled processes communicating with each other using language-agnostic application programming interfaces (API). Microservice approaches and container-based operating system virtualization experience a renaissance in cloud computing. Especially container-based virtualization approaches are often mentioned to be a high-performance alternative to hypervisors [3]. *Docker* [4] is such a container solution, and it is based on operating system virtualization using Linux containers. Recent performance studies show only little performance impacts to processing, memory, network or I/O [5]. That is why *Docker* proclaims itself a ”lightweight virtualization platform” providing a standard runtime, image format, and build system for Linux containers deployable to any Infrastructure as a Service (IaaS) environment.

This study investigated the performance impact of Linux containers on top of hypervisor based virtual machines logically connected by an (encrypted) overlay network. This is a common use case in IaaS Cloud Computing being applied by popular microservice platforms like Mesos [6], CoreOS [7] or Kubernetes [8] (the reader may want to study a detailed analysis of such kind of platforms [9]). Nevertheless, corresponding performance impacts have been hardly investigated so far. Distributed cloud based microservice systems of typical complexity often use hypertext transfer protocol (HTTP) based and representational state transfer (REST) styled protocols to enable horizontally scalable system designs [10]. If these systems are deployed in public clouds, additional requirements for encrypted data transfer arise. There exist several open source projects providing such a microservice approach on top of IaaS provider specific infrastructures using this approach (e.g. Mesos, Kubernetes, CoreOS and more). These approaches are intended to be deployable to public or private IaaS infrastructures [9]. So in fact, these approaches apply operating system virtualization (containers) on top of hypervisors (IaaS infrastructures). Although almost all of these microservice frameworks rely heavily on the combination of containerization on top of hypervisors, and some of these approaches introduce additional overlay networking and data encryption layers, corresponding performance impacts have been hardly analyzed so far. Most performance studies compare container performance with virtual machine performance but not container performance on top of virtual machines (see Felter at al. [5] for a typical performance study).

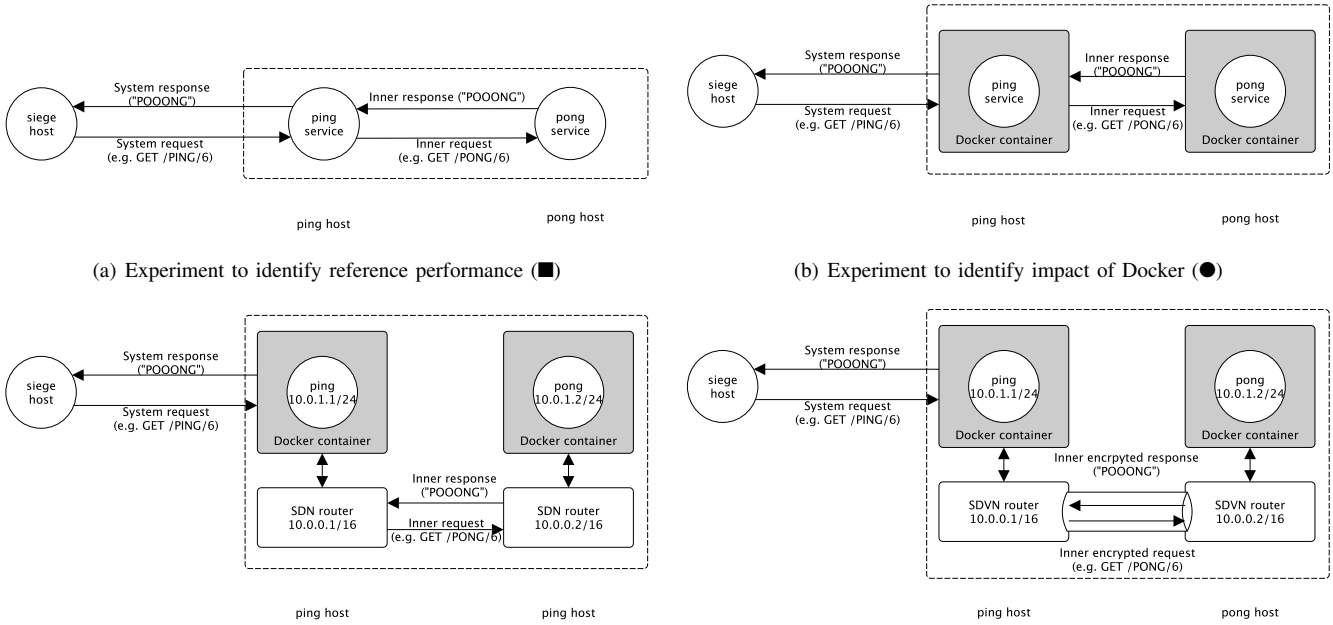
Because overlay networks are often reduced to distributed hashtable (DHT) or peer-to-peer approaches, this paper uses the term software defined virtual networks (SDVN). SDVNs, in the understanding of this paper, are used to provide a logical internet protocol (IP) network for containers on top of IaaS infrastructures.

Section II presents related work about state-of-the-art container approaches and SDVN solutions. Section III explains the experiment design to identify performance impacts of containers, SDVNs and encryption. The benchmark tooling [11] and the performance data collected is provided online [12]. Resulting performance impacts are discussed in Section IV. Derived design recommendations to minimize performance impacts on application, overlay network and IaaS infrastructure layer are presented in concluding Section V.

II. RELATED WORK

Although container based operating system virtualization is postulated to be a scalable and high-performance alternative to hypervisors, hypervisors are the standard approach for IaaS cloud computing [3]. Felter et al. provided a very detailed analysis on CPU, memory, storage and networking resources to explore the performance of traditional virtual machine deployments, and contrast them with the use of Linux containers provided via *Docker* [5]. Their results indicate that benchmarks that have been run in a *Docker* container,

 ISBN: 978-1-61208-388-9



(a) Experiment to identify reference performance (■)

(b) Experiment to identify impact of Docker (●)

(c) Experiment to identify impact of SDVN (▲)

(d) Experiment to identify impact of encryption (▼)

Figure 1. Experiments

show almost the same performance (floating point processing, memory transfers, network bandwidth and latencies, block I/O and database performances) like benchmarks run on ”bare metal” systems. Nevertheless, Felter et al. did not analyze the impact of containers on top of hypervisors.

Although there exist several SDVN solutions for *Docker*, only one open source based SDVN has been identified, which is able to encrypt underlying data transfers: *Weave* [13]. That is why other SDVN approaches for *Docker* like *flannel* [14] or *docknet* [15] are not covered by this study. Pure virtual local area network (VLAN) solutions like *Open vSwitch (OVS)* [16] are not considered, because *OVS* is not to be designed for operating system virtualization. So, *weave* remained as the only appropriate SDVN candidate for this study. But the author is confident that this will change in the future and more encryptable SDVN solutions will arise.

Weave creates a network bridge on *Docker* hosts to enable SDVN for *Docker* containers. Each container on a host is connected to that bridge. A *weave* router captures Ethernet packets from its bridge-connected interface in promiscuous mode. Captured packets are forwarded over the user datagram protocol (UDP) to *weave* router peers running on other hosts. These UDP ”connections” are duplex, can traverse firewalls and can be encrypted.

To analyze the performance impact of containers, software defined networks and encryption, this paper considered several contributions on cloud related network performance analysis (see [17], [18], [19], [20], [21], [22]). But none of these contributions focused explicitly on horizontally scalable systems with HTTP-based and REST-like protocols. To address this common use case for microservice architectures, this paper proposes the following experiment design.

III. EXPERIMENT DESIGN

This study analyzed the network performance impact of container, SDVN and encryption layers on the performance impact of distributed cloud based systems using HTTP-based REST-based protocols. Therefore, five experiments have been designed (see Figure 1). The analyzed *ping-pong* system relied on a REST-like and HTTP-based protocol to exchange data. Apachebench [23] was used to collect performance data of the *ping-pong* system. *Siege*, *ping* and *pong* servers have been deployed to the Amazon Web Services (AWS) IaaS infrastructure on a m3.medium instance type. Experiments have been run in eu-west-1c availability zone (Ireland). The *siege* server run the apachebench benchmark. The *ping* and *pong* application were developed using Googles Dart programming language [24]. To understand the performance impact of containers, SDVN and encryption to network performance, the study analyzed the data transfer rate $trans(m)$ of m byte long messages.

■ The **reference experiment** shown in Figure 1(a), was used to collect reference performance data of the *ping-pong* system deployed to different virtual machines interacting with a REST-like and HTTP based protocol. No containers, SDVN or encryption were used in this experiment. Further experiments added a container, a SDVN and an encryption layer to measure their impact on network performance. A *ping* host interacts with a *pong* host to provide its service. Whenever the *ping* host is requested by *siege* host, the *ping* host relays the original request to the *pong* host. The *pong* host answers the request with a response message. The *siege* host can define the inner message and system response message length by query. All requests are performed using the HTTP protocol. The *siege* host is used to run several *apachebench* benchmark runs with increasing requested message sizes to measure the system performance of the *ping-pong* system. This paper refers

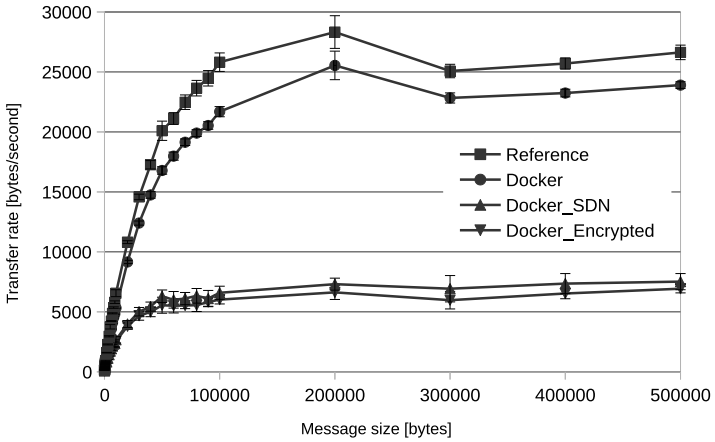


Figure 2. Absolute performance impact on transfer rates

to measured transfer rates for a message size of m (bytes) of this experiment as $trans_{\square}(m)$. These absolute values are presented in Figure 2.

● The intent of the **Docker experiment** was to figure out the impact of an additional container layer to network performance (Figure 1(b)). So, the *ping* and *pong* services are provided as containers to add an additional container layer to the reference experiment. Every performance impact must be due to this container layer. The measured transfer rates are denominated as $trans_{\bigcirc}(m)$. The impact of containers on transfer rates is calculated as follows and presented in Figure 3(a):

$$\bigcirc_{trans}(m) = \frac{trans_{\bigcirc}(m)}{trans_{\square}(m)} \tag{1}$$

▲ The intent of the **SDVN experiment** shown in Figure 1(c) was to figure out the impact of an additional SDVN layer to network performance. This experiment connects *ping* and *pong* containers by a SDVN. So, every data transfer must pass the SDVN solution between *ping* and *pong*. This paper refers to measured transfer rates for a message size of m (bytes) of this experiment as $trans_{\Delta}(m)$. The impact of SDVN on transfer rates for a message size m is calculated as follows and presented in Figure 3(a):

$$\Delta trans(m) = \frac{trans_{\Delta}(m)}{trans_{\square}(m)} \tag{2}$$

▼ The **encryption experiment** (see Figure 1(d)) figured out the impact of an additional data encryption layer on top of a SDVN layer. Additionally, this experiment encrypts the SDVN network. Corresponding measured transfer rates are denominated as $trans_{\nabla}(m)$. The impact of SDVN on transfer rates for a message size m is calculated as follows and is presented in Figure 3(a):

$$\nabla trans(m) = \frac{trans_{\nabla}(m)}{trans_{\square}(m)} \tag{3}$$

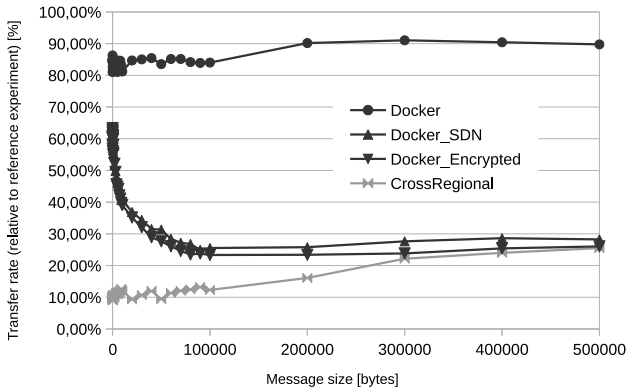
⋈ The intent of **cross-regional experiment** was to figure out the impact of an cross-regional deployment to network performance. Although this was not the main focus of the study, this use case has been taken into consideration to generate a more graspable performance impact understanding for the reader. The setting has been the same as in Figure 1(a), except that the *ping* and *pong* hosts were deployed to different regions of the AWS infrastructure. *ping* (and the *siege* host) were deployed to the AWS region eu-west-1c (EU, Ireland) and the *pong* host was deployed to AWS region ap-northeast-1c (Japan, Tokyo). Data from this experiment is only used to compare container, SDVN and encrypted SDVN performance with a cross-regional performance impact in a qualitative manner. A cross-regional impact might be more intuitively graspable for the reader. The cross-regional impact on transfer rates is presented in Figure 3(a) as a lightgrey line.

IV. DISCUSSION OF RESULTS

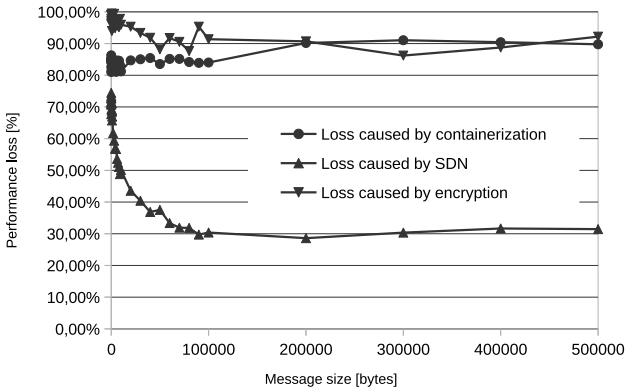
The results presented are based on more than 12 hours of benchmark runs, which resulted in a transfer of over 316GB of data requested by more than 6 million HTTP requests (see Table I). Reference and Docker experiments show only minor standard deviations. The maximum deviations were measured for very small message sizes (10 and 20 byte). Standard deviations increased with the introduction of SDVN. So, the collected deviation data indicates that the experiment setting produces reliable data (increasing deviations of SDVN experiment have to do with the technical impacts of SDVNs, they are not due to experiment design and will be discussed by this paper).

● *Container impact:* Containers are stated to be lightweight and to have only negligible performance impacts [5]. The *Docker* experiment shows a somewhat different picture. A non negligible performance loss can be identified for data transfer rates (see Figure 2). An additional container layer on top of a bare virtual machine reduces the performance to 80% (for message sizes smaller than 100 kBytes) to 90% (for message sizes greater than 100kBytes). The study also included a cross-zone deployment (similar to the cross-region deployment but in two different availability zones of the same AWS region). It turned out that a cross-zone deployment

 ISBN: 978-1-61208-388-9



(a) comparison of transfer rates (100% means no loss)



(b) Resulting transfer losses (100% means no loss)

Figure 3. Relative comparison of performance indicators

shows almost the same performance like an one-zone deployment (reference experiment). Containers show a significant higher performance impact than cross-zone deployments in IaaS clouds. So, compared with cross-zone deployments (non-measurable effects), we have to say that containers have measurable (non-negligible) impacts to network performance.

▲ *SDVN impact:* The impact of analyzed SDVN solution *weave* reduces data transfer rates from 25 kB/s to about 7,5kB/s (see Figure 2). Figure 3a shows the relative performance of the SDVN experiment. SDVN experiments show only 60% performance of the reference experiment for small message sizes going down to about 25% performance for message sizes greater than 100kB. The SDVN experiment shows a comparable performance impact like a cross-regional deployment (for big message sizes, Ireland ↔ Japan). *Weave* SDVN routers are provided as *Docker* containers. The SDVN experiment measures the performance impact of containerization **and** SDVN. To identify the pure SDVN effect, the reader has to compare SDVN data (Δ) relative to the performance data of containers ($\bigcirc$) to exclude the container effects (see Figure 3b).

$$impact_{\Delta}(m) = \frac{trans_{\Delta}(m)}{trans_{\bigcirc}(m)} \tag{4}$$

To avoid container losses, SDVN solutions should be provided directly on the host and not in a containerized form. This should reduce the performance loss about 10% to 20%. Furthermore, it is noted that tests were running on a single-core virtual machine (m3.medium type). In saturated network load situations, the *weave* router contends for CPU with the application processes, so it will saturate faster compared with Reference or *Docker* experiment where the network is handled by the hypervisor and physical network, outside of such contention. This effect explains the severe performance impacts shown in Figure 3. That lead us to the design conclusion that SDVN solutions should always run on multi-core systems to avoid severe performance impacts due to contention.

▼ *Encryption impact:* Additional encryption shows only minor impacts to transfer rates compared with SDVN without encryption (see Figure 2). So, most of the performance losses are due to SDVN and not because of encryption. To identify the pure encryption effect, encrypted SDVN data (∇) has to be compared relative to the performance data of SDVN experiment (Δ).

TABLE I. RELATIVE STANDARD DEVIATIONS OF MEASURED TRANSFER RATES

		%RSD		
Experiment	**Data**	**Min**	**Avg**	**Max**
Reference	90 GB	0,9	2,9	15,9
Cross Regional	19 GB	0,9	14,9	28,7
Docker	57 GB	0,8	2,0	10,3
Docker_SDVN	75 GB	0,4	11,3	21,2
Docker_Encrypted	75 GB	0,5	10,9	16,2

$$impact_{\nabla}(m) = \frac{trans_{\nabla}(m)}{trans_{\Delta}(m)} \tag{5}$$

Encryption reduces the transfer performance down to about 90% compared with the transfer rates of non encrypted data transfers. For smaller message sizes this negative effect of encryption gets even more and more negligible. In other words, especially for small message sizes encryption is not a substantial performance killer compared to SDVN impact (see Figure 3b). For bigger message sizes the data transfer performance impact of encryption is comparable to containerization.

V. CONCLUSION

This study analyzed performance impact of containers, overlay networks and encryption to overall network performance of HTTP-based and REST-like services deployed to IaaS cloud infrastructures. Obviously, our conclusions should be cross checked with other SDVN solutions for containers. The provided data [12] and benchmarking tools to apply the presented methodology [11] can be used as benchmark for that purpose. Nevertheless, some of the study results can be used to derive some design recommendations for cloud deployed HTTP-based and REST-like systems of general applicability.

Although **containers** are stated to be lightweight [3] [5], this study shows that container impact on network performance

 ISBN: 978-1-61208-388-9

is not negligible. Containers show a performance impact of about 10% to 20%. The impact of **overlay networks** can be even worse. The analyzed SDVN solution showed a performance impact of about 30% to 70%, which is comparable to a cross regional deployment of a service between Ireland and Japan. **Encryption** performance loss is minor, especially for small message sizes.

The results show that performance impacts of overlay networks can be minimized on several layers. On **application layer** message sizes between system components should be minimized whenever possible. Network performance impact gets worse with increasing message sizes. On **overlay network layer** performance could be optimized by 10% to 20% by providing SDVN router applications directly on the host (in a not containerized form, because 10% to 20% are due to general container losses). On **infrastructure layer**, the SDVN routers should be deployed to multi core virtual machines to avoid situations, where SDVN routers contend for CPU with application processes.

So containers, which are often mentioned to be lightweight, are not lightweight under all circumstances. Nevertheless, the reader should not conclude to avoid container and SDVN technologies in general. Container and SDVN technologies provide more flexibility and manageability in designing complex horizontally scalable distributed cloud systems. And there is nothing wrong about flexibility and manageability of complex systems. That is why container solutions like Docker and microservice approaches regain so much attention recently. But container and SDVN technologies should be always used with above mentioned performance implications in mind.

ACKNOWLEDGMENT

This study was funded by German Federal Ministry of Education and Research (Project Cloud TRANSIT, 03FH021PX4). The author thanks Lübeck University (Institute of Telematics) and fat IT solution GmbH (Kiel) for their support of Cloud TRANSIT. The author also thanks Bryan Boreham of zett.io for checking our data of zett.io's *weave* solution (which might show now better results than the analyzed first version of *weave*).

REFERENCES

[1] M. Fowler and J. Lewis. Microservices. Last access 17th Feb. 2015. [Online]. Available: http://martinfowler.com/articles/microservices.html [retrieved: March, 2014]

[2] S. Newman, Building Microservices. O'Reilly and Associates, 2015.

[3] S. Soltesz, H. Pötzl, M. E. Fiuczynski, A. Bavier, and L. Peterson, "Container-based operating system virtualization: A scalable, high-performance alternative to hypervisors," SIGOPS Oper. Syst. Rev., vol. 41, no. 3, Mar. 2007, pp. 275–287.

[4] Docker. Last access 17th Feb. 2015. [Online]. Available: https://docker.com

[5] W. Felter, A. Ferreira, R. Rajamony, and J. Rubio, "An updated performance comparison of virtual machines and linux containers," IBM Research Division, Austin Research Laboratory, Tech. Rep., 2014.

[6] Mesos. Last access 17th Feb. 2015. [Online]. Available: https://mesos.apache.org

[7] Coreos. Last access 17th Feb. 2015. [Online]. Available: https://coreos.com

[8] Kubernetes. Last access 17th Feb. 2015. [Online]. Available: https://github.com/GoogleCloudPlatform/kubernetes

[9] N. Kratzke, "A lightweight virtualization cluster reference architecture derived from open source paas platforms," Open Journal of Mobile Computing and Cloud Computing (MCCC), vol. 1, no. 2, 2014, pp. 17–30.

[10] R. T. Fielding, "Architectural styles and the design of network-based software architectures," Ph.D. dissertation, University of California, Irvine, 2000.

[11] N. Kratzke. Ping pong - a distributed http-based and rest-like ping-pong system for test and benchmarking purposes. Last access 17th Feb. 2015. [Online]. Available: https://github.com/nkratzke/pingpong

[12] ——. Collected performance data. Last access 17th Feb. 2015. [Online]. Available: https://github.com/nkratzke/sdvn-impact-database

[13] Weave. Last access 17th Feb. 2015. [Online]. Available: https://github.com/zettio/weave

[14] Flannel. Last access 17th Feb. 2015. [Online]. Available: https://github.com/coreos/flannel

[15] Docknet. Last access 17th Feb. 2015. [Online]. Available: https://github.com/helander/docknet

[16] Open vswitch. Last access 17th Feb. 2015. [Online]. Available: http://openvswitch.org

[17] D. Mosberger and T. Jin, "Httperf—a tool for measuring web server performance," SIGMETRICS Perform. Eval. Rev., vol. 26, no. 3, Dec. 1998, pp. 31–37. [Online]. Available: http://doi.acm.org/10.1145/306225.306235

[18] G. Memik, W. H. Mangione-Smith, and W. Hu, "Netbench: A benchmarking suite for network processors," in Proceedings of the 2001 IEEE/ACM International Conference on Computer-aided Design, ser. ICCAD '01. Piscataway, NJ, USA: IEEE Press, 2001, pp. 39–42.

[19] J. Verdú, J. Garcí, M. Nemirovsky, and M. Valero, "Architectural impact of stateful networking applications," in Proceedings of the 2005 ACM Symposium on Architecture for Networking and Communications Systems, ser. ANCS '05. New York, NY, USA: ACM, 2005, pp. 11–18.

[20] K. Jackson, L. Ramakrishnan, K. Muriki, S. Canon, S. Cholia, J. Shalf, H. J. Wasserman, and N. Wright, "Performance analysis of high performance computing applications on the amazon web services cloud," in Cloud Computing Technology and Science (CloudCom), 2010 IEEE Second International Conference on, Nov 2010, pp. 159–168.

[21] G. Wang and T. Ng, "The impact of virtualization on network performance of amazon ec2 data center," in INFOCOM, 2010 Proceedings IEEE, March 2010, pp. 1–9.

[22] D. Jayasinghe, S. Malkowski, J. LI, Q. Wang, Z. Wang, and C. Pu, "Variations in performance and scalability: An experimental study in iaas clouds using multi-tier workloads," Services Computing, IEEE Transactions on, vol. 7, no. 2, April 2014, pp. 293–306.

[23] Apachebench. Last access 17th Feb. 2015. [Online]. Available: http://httpd.apache.org/docs/2.2/programs/ab.html

[24] Dart. Last access 17th Feb. 2015. [Online]. Available: https://www.dartlang.org/

 ISBN: 978-1-61208-388-9