Survey on Performance Models of Container Networks

David de Andrés Hernández
Department of Electrical and Computer Engineering
Technical University of Munich
Email: deandres.hernandez@tum.de

***Abstract*—Reliability, scalability, and flexibility are some of the benefits of cloud architectures. When *cloudifying* any system, the choice of an appropriate container network solution is critical. The wrong container network choice can turn a working system unviable. This translates into performance degradation, security exposures or unmanageable complexity. Yet the number of solutions targeting the cloud environment is vast and the angles approaching the challenges are varied. Therefore, understanding and evaluating the benefits and bottlenecks of the available solutions and technologies is a tedious but crucial process. To be thorough, this evaluation requires of several steps. A mandatory first step is to identify the requirements of the system to be *cloudified*. Further, consulting benchmarks and models can give an indication of viability. In the last stages, a high-fidelity model can precisely simulate the results that different configurations and changes can have in the system's behaviour.**

This paper gives an overview of different container solutions, performance benchmarks and performance models. Finally, the major factors influencing container networks are conceptually introduced.

I. Introduction

Cloud architectures are motivated by both economies of scale and resources optimization and are enabled by virtualization technologies. The cloud promises elasticity, scalability, reliability, availability, and increased operability. However, performance degradation and reduced portability are often the price of virtualization. To mitigate this effect, containers, a form of lightweight virtualization, emerged. Thanks to their alternative means of partitioning resources, the virtualization overhead is drastically reduced, and the deployment process is expedited. These benefits favoured the standardization of containers and that in turn made portability an additional strength of this technology.

Microservices architectures exploit containers even further. This architecture borrows the encapsulation principle of software development and breaks applications into stand-alone containers. In this approach, each container is only responsible for a single function. This function decoupling means that components can be updated or replaced without affecting the remaining. Moreover, applications can be granularly scaled by load-balancing a function into several instances of the same container. If we take a step further and make individual container instances stateless, containers can be re-spawned, if necessary, without impact to the supported application. The price to pay for microservices is the need of automation and orchestration software to overcome the explosion in the number of components. Overall, we can see that microservices and cloud principles are appropriately aligned.

We can derive that network connectivity among containers is paramount for their correct functioning. However, it is not free of challenges. A microservice-populated cloud changes constantly within seconds and containers can be considered ubiquitous. All possible traffic patterns take place: between containers in the same VM, between containers in different VMs, between containers in different hosts, between containers in different datacenters, and many more. In this scenario, the host's OS becomes an important element of the networking infrastructure of the cloud. With one exception, it was not conceived for such scale. For this reason, the networking performance of containers must be carefully considered as it can a turn working system unviable when migrated to cloud architectures. The main cause of this degradation is due to overhead processing of packets through the OS's networking stack. Another important factor is that caused by the processing of the headers of overlay networks. To enable the communication of containers in constant move, complex overlay networks are required. But again, the host's OS where not conceived to process these headers at this scale efficiently. To overcome the performance challenge operators have conceived sophisticated frameworks for high-performance packet IO.

Cloud networks must also remain extremely flexible as they must often support multi-tenancy while remaining secure.

Due to the immense number of choices for container networking solutions, evaluating the application needs as well as the benefits and bottlenecks of the available solutions is of vital importance. This survey gives an overview of different container solutions and their underlying technologies. In addition, it presents a collection of available performance benchmarks and models as well as the environments studied so far. Finally, the major factors influencing container networks are conceptually introduced.

II. Background

A. Subsection Heading Here

Subsection text here.

1) Subsubsection Heading Here: Subsubsection text here.

III. Current Solutions

IV. Benchmarks

V. Models

VI. Factors Analysis

VII. Conclusion

The conclusion goes here.

Acknowledgment

The authors would like to thank...

References

[1] H. Kopka and P. W. Daly, *A Guide to LaTeX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.