Quantifying the Isolation Characteristics in Container Environments

Chang Zhao[1], Yusen Wu[1], Zujie Ren[1(✉)], Weisong Shi[2], Yongjian Ren[1], and Jian Wan[3]

[1] Hangzhou Dianzi University, Hangzhou, China
chaos_zhch@163.com, yusen.wu08@gmail.com, renzj@hdu.edu.cn, yongjian.ren@infocore.cn
[2] Wayne State University, Detroit, USA
weisong@wayne.edu
[3] Zhejiang University of Science and Technology, Hangzhou, China
wanjian@zust.edu.cn

Abstract. In recent years, container technologies have attracted intensive attention due to the features of light-weight and easy-portability. The performance isolation between containers is becoming a significant challenge, especially in terms of network throughput and disk I/O. In traditional VM environments, the performance isolation is often calculated based on performance loss ratio. In container environments, the performance loss of well-behaved containers may be incurred not only by misbehaving containers but also by container orchestration and management. Therefore, the measurement models that only take performance loss into consideration will be not accurate enough. In this paper, we proposed a novel performance isolation measurement model that combines the performance loss and resource shrinkage of containers. Experimental results validate the effectiveness of our proposed model. Our results highlight the performance isolation between containers is different with the issue in VM environments.

Keywords: Containers · Performance isolation · Isolation measurement models

1 Introduction

Containers enable new ways to run applications by containerizing applications and services, making them portable, extensible, and easy to be transferred between private data centers and public clouds. However, containers suffer from a poor performance isolation as they share both OS kernels and physical servers. And in container environments, the life cycle and resources of containers are controlled by the container orchestrations. Therefore, the performance isolation measurement model in VM environments, is inapplicable for container environments.

X. Shi et al. (Eds.): NPC 2017, LNCS 10578, pp. 145–149, 2017.
DOI: 10.1007/978-3-319-68210-5_17

In this paper, we proposed a comprehensive performance isolation measurement model that combines the performance loss and resource shrinkage of containers. The advantage of this model is that if the resource occupied by each container varies, the model can express the resource change, as well as the performance change. We conducted a group of performance evaluation experiments to validate the effectiveness of our proposed model.

2 Related Work

Performance isolation is one of the desirable features in virtualized environments [1]. In the field of traditional VMs, many research effort has been conducted to improve the performance isolation between VMs. Gupta *et al.* [2] developed a XEN-based monitoring system to monitor CPU utilizations of each VM, and dynamically schedule the resource allocation of CPU shares to enhance isolation. Shi *et al.* [3] proposed in a smart EdgeOS, the performance isolation might be more complicated than which in a distributed system. Therefore, a well designed control access mechanism should be added to the service management layer in the EdgeOS.

We noticed a few research works on performance isolation measured designed for traditional virtual machines [4]. These works often use the performance loss ratio to measure the isolation, which works on the assumption that the resource capacity of each virtualized machine is static. While for containers, the cause for performance losses for a container may not only due to the interference by misbehaving containers, but also the decrease of resources. Therefore, the existing models for VMs are inapplicable in container environments.

3 Performance Isolation Model

3.1 A Model Combining Performance Loss and Resource Shrinkage

Suppose there are two servers with same capacity, S_1 and S_2. Both of them host two containers, S_1 hosts container A_1 and B_1, and S_2 hosts containers A_2 and B_2. The configurations of all the four containers are same. Both B_1 and B_2 are misbehaving containers. Suppose A_1 and A_2 suffer same performance loss ratio, such as twenty percent. Meanwhile, the resource consumed by A_1 decreases a considerable portion, and A_2 does not. The isolation of S_1 should be better than the one of the S_2, because A_1 consume less resources to achieve a specific performance.

As shown in the following model 1, the performance isolation is associated with P_{loss} and R_{skg}. P_{loss} represents the performance loss degree of the containers, R_{skg} represents the resource shrinkage degree of the containers, and the larger value of P_{loss} and R_{skg} means the slower drop of the corresponding metrics. Through this model 1, we can normalize the performance isolation degree to the range of $[0, 1]$.

$$I = f(P_{loss}, R_{skg}) = \frac{(1+\beta^2) * P_{loss} * (1 - R_{skg})}{\beta^2 P_{loss} + (1 - R_{skg})} \tag{1}$$

The β in the formula 1 represents the scale factor used to adjust the weight of performance loss degree and resource shrinkage degree. A large β means that the performance loss is emphasized, while a small β means the resource elasticness is emphasized. By default, we can set β equal to 1.

3.2 Quantifying Performance Loss

In a well-isolated container system, the performance increasing of misbehaving containers does not affect the performance of other containers, as the green line shown in Fig. 1. In a no isolation container system, as the performance of misbehaving containers increasing, the performance of the well-behaving containers will drop linearly as the red line shown. And performance increasing of misbehaving containers is the same as the performance loss of the well-behaving containers.

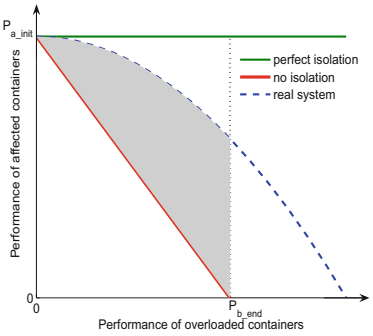

Fig. 1. Isolation curve including perfect isolation, no isolation and real system isolation. (Color figure online)

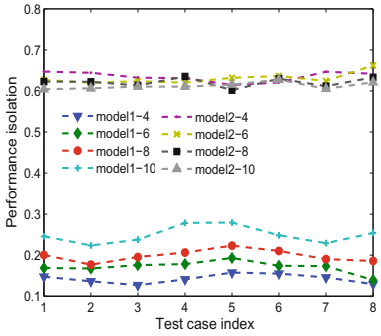

Fig. 2. Performance isolation with different workload of well-behaving containers.

However, in an actual container system, the performance of the well-behaving containers shows a downward trend in the curve, as the blue dotted line in Fig. 1. Thus we found that the blue dotted line closer to the green line and the shadow area is larger, the performance loss degree is better. We use the function $P_a = f(P_b)$ to represent the blue curve, then we define the calculation formula of the P_{loss} as the formula 2. For the model 1, we can also use the formula 2 to calculate R_{loss}.

$$I = \frac{\int_0^{P_{b\_end}} f(P_b)dP_b - P_{b\_end} * P_{a\_init}/2}{P_{b\_end} * P_{a\_init}/2} \tag{2}$$

4 Validation of the Model

4.1 Methodologies

In the experiment, well-behaving containers run SQL operations with a stable rate, while the workloads in misbehaving containers are gradually increased. To put workloads on these containers, we deploy MySQL database and use Sysbench benchmarks to generate SQL workloads.

4.2 Validation and Model Comparison

The workload in well-behaving containers have great influence on the performance isolation. When the workload on well-behaving containers is low, misbehaving containers can preempt the resources from well-behaving containers and impact the performance of well-behaving containers. In this case, the performance interference between containers is obvious and the performance isolation is poor.

The experiment controls the workload of the containers by tuning the number of threads of Sysbench. When the number of threads increases, the workload on the containers increases. The number of threads in the experiment was set to 4, 6, 8 and 10, respectively. We use the performance loss ratio and our measurement model to calculate the performance isolation, and then the experimental results were compared. In each case, eight experiments were executed and 64 sets of performance statistics were collected.

Experimental results are shown in Fig. 2. Model1 represents our proposed model, and model2 represents the traditional model that only considers performance loss ratio. Model-4 means the result of 4 threads in our model. When the workload in well-behaving containers changes, it is hard to accurately reflect the changes of the performance isolation. While for our model, the results for each group present obvious intervals, and demonstrate the distinct degrees of the performance isolation under the different workloads.

5 Summary

In this work, we presented an early-stage research work on the isolation in containers. We hope this work can motivate the container community to further address some open problems, such as resource scheduling among containers and workload-aware container orchestration. For more details, please refer to the extend version of this work [5].

Acknowledgement. This research is supported by the NSF of China (No. 61572163). Weisong Shi is in part supported by National Science Foundation (NSF) grant CNS-1563728.

References

1. Krebs, R.: Performance Isolation in Multi-Tenant Applications. PhD thesis, Karlsruhe Institute of Technology (2015)
2. Gupta, D., Cherkasova, L., Gardner, R., Vahdat, A.: Enforcing performance isolation across virtual machines in Xen. In: van Steen, M., Henning, M. (eds.) Middleware 2006. LNCS, vol. 4290, pp. 342–362. Springer, Heidelberg (2006). doi:10.1007/11925071_18
3. Shi, W., Cao, J., Zhang, Q., Li, Y., Xu, L.: Edge computing: vision and challenges. IEEE Internet Things J. **3**(5), 637–646 (2016)
4. Walraven, S., Monheim, T., Truyen, E., Joosen, W.: Towards performance isolation in multi-tenant SAAS applications. In: Proceedings of the 7th Workshop on Middleware for Next Generation Internet Computing, p. 6. ACM (2012)
5. Zhao, C., Wu, Y., Ren, Z., Shi, W., Ren, Y., Wan, J.: Quantifying the isolation characteristics in container environments. Technical report No. MIST-TR-2017-010, Wayne State University (2017). http://www.cs.wayne.edu/~weisong/papers/MIST-TR-2017-010.pdf