An Empirical Study on the Impact of Selected Host Configuration Parameters on Container Start Times

Martin Straesser, Nicholas Erhard, Samuel Kounev
University of Würzburg, Germany

Abstract

Modern cloud technologies rely on container virtualization as the basis for application deployment as they show faster start times compared to virtual machines. Understanding the impacting factors for container start times is crucial for application developers and platform engineers to identify optimization potential. In this work, we extend our previous empirical study on container start times by analyzing the impact of selected host configuration parameters (e.g., disk type, RAM size, CPU vendor). Our insights include that the disk type is the most important platform parameter analyzed. However, attributes of the container image and the host machine must always be considered together to assess container start times accurately.

1 Introduction

Containers have become an integral part of modern software development and cloud computing. The market size for application containers is expected to exceed 19 billion USD in 2029.[1] One of the major advantages of containers over virtual machines is their reduced start time. Low start times not only enable the use of the latest cloud technologies such as serverless computing but are also an active field of research [2, 3, 4].

In an earlier empirical study, we focused on variations in start times between different containers. Understanding the factors that influence container start times enables the identification of potential optimization targets for application developers and platform engineers. In [3], we analyzed how image configuration parameters such as the image size and number of root file system layers impact container start times. We also noted that the start time is heavily influenced by the test machine on which we run the containers.

In this study, we extend our previous study by running 1008 different container images on test machines that differ by CPU, RAM, disk, and OS specifications. Our study is based on a Plackett-Burman experimental design and includes over 160,000 container start measurements. Our main research questions are: (1) To what extent do container start times vary on different host configurations? (2) Which platform parameters impact container start times most, and what is the relationship to image configuration parameters?

2 Foundations and Previous Work

The start time of a container is part of its readiness time. The readiness time is the time a container needs to become ready, which means for web applications to be able to serve user requests. The readiness time includes the image download (pull), the container start, and an application-specific setup (e.g., loading a Python runtime). The readiness time can only be measured if we know the container's intended function, as we have to define a readiness probe (health check). In contrast, the start time can be investigated without knowing the container internals and external dependencies, as the start process of the container is content-agnostic, according to the OCI specification.[2]

In our previous work, we conducted an empirical study on container start times and investigated which parameters from the OCI image configuration impact the start time most. Therefore, we created a large dataset from the public image repository Docker Hub containing more than 200,000 open-source container images. The dataset contains images with various characteristics (e.g., the container size ranges from a few bytes to over 90GB). Because of time and cost constraints, we could not test all of the over 200,000 images; instead, we used stratified sampling to retrieve a diverse, representative sample. While our previous work focused on the container image, we also conducted measurements in two different environments: Google Cloud Platform (GCP) `e2-medium` virtual machines and a self-hosted testbed.

3 Study Design

The results from the previous study indicate that, besides image configuration parameters, platform parameters of the test machines also influence the container start time. Because of the high diversity of possible host configurations, we cannot conduct a representative study of all platform parameters. Although the results might not be universally applicable, we can still point out tendencies and test our presumptions from previous results, for example, that the

[1] `https://www.mordorintelligence.com/industry-reports/application-container-market`

[2] `https://github.com/opencontainers/runtime-spec`

Parameter	Low (-1)	High (+1)
Disk	Standard	SSD
RAM	4GB	16GB
OS	Ubuntu 22.04	Fedora Cloud Base 36
Control Groups	v1	v2
CPU Vendor	AMD Milan	Intel Cascade Lake
CPU Cores	1 (2 vCPUs)	4 (8 vCPUs)

Table 1: Encoding platform parameters to Plackett-Burman factors.

disk type (HDD or SSD) has a significant impact. For this purpose, we use different configurations of GCP `n2-custom` and `n2d-custom` VMs and start the same container images on different machine types. We use the popular container technology stack consisting of Docker, containerd, and the OCI reference runtime runc for executing containers. Our methodology is based on the Plackett-Burman experimental design [1], which reduces the number of experiments to conduct compared to a full factorial analysis.

The Plackett-Burman experimental design is used to investigate the impact of m controllable parameters on a response measure (in our case, container start time). Each parameter has two alternatives, which are encoded as -1 (*low*) and 1 (*high*). In the following, we explain the parameters considered in this section in more detail and which values we have chosen for *low* and *high*. Table 1 shows an overview of the tested parameters. The host parameters are selected based on the settings that can be set when creating a virtual machine on GCP. An essential parameter is the *disk type*. GCP offers standard (HDD-backed), balanced, and SSD (both SSD-backed) disks for virtual machines. SSD disks offer significantly higher read and write throughput than standard disks. In our experiment design, we assign the value of -1 (*low*) to standard disks and 1 (*high*) to SSD disks. As the second parameter, we consider the *RAM size*. We choose 4GB for *low* and 16GB for *high*. For the CPUs, we consider two parameters. We distinguish Intel Cascade Lake and AMD Milan processors as *CPU vendors*. We encode the Intel processor as 1 (*high*) because it has a higher base frequency than the AMD processor (2.8 vs. 2.45 GHz).

In addition to the hardware configuration of the VMs, we also consider two software-related settings. As *operating systems*, we choose Fedora Cloud Base 36 and Ubuntu 22.04 LTS. Fedora tends to use the latest software, while Ubuntu focuses more on stability. The Fedora OS uses a newer Linux kernel (version 5.17) than Ubuntu (version 5.15). Hence, we encode Fedora as *high* and Ubuntu as *low*. While it is out of scope to examine impacts from all software dependencies present on the host, we take a closer look at *cgroup versions*. Linux containers are based on so-called control groups, and the kernel offers two major versions of control groups. Version 2 is the newer version and promises more functionalities and increased

Config No.	Disk	RAM	OS	cgroups	CPU Cores	CPU Vendor
1	+1	+1	+1	-1	+1	-1
2	-1	+1	+1	+1	-1	+1
3	-1	-1	+1	+1	+1	-1
4	+1	-1	-1	+1	+1	+1
5	-1	+1	-1	-1	+1	+1
6	+1	-1	+1	-1	-1	+1
7	+1	+1	-1	+1	-1	-1
8	-1	-1	-1	-1	-1	-1

Table 2: Plackett-Burman design matrix.

Config No.	Start time statistics [ms]				
	Min	Median	Mean	P95	Max
1	234	453	1370	4881	20221
2	245	2389	18865	98073	363794
3	227	9364	12366	28751	121088
4	232	1098	991	1367	5555
5	233	25776	40851	124614	354270
6	239	521	788	1525	5764
7	240	1878	2390	5993	14570
8	227	14169	15528	31403	217957

Table 3: Start time statistics for different host configurations.

performance compared to version 1. In our experimental design, we encode `cgroups` v2 as *high* and v1 as *low*.

Similar to our previous work, we use a sample of 1008 images created with the same methodology described in [3]. The Plackett-Burman design specifies which parameter combinations are to be tested, as Table 2 shows. Each of our 1008 sample images is run 20 times on each of the eight host configurations. This results in a total of 161,280 container start measurements considered in this work.

4 Results

We first report a comprehensible summary of our measurement results. Table 3 shows statistical measures of the start time measurements for each tested host configuration. The lowest start time is below 250 ms for all configurations. Significant differences between host configurations are visible in all other reported measures. For example, the configurations differ by up to two orders of magnitude in the mean and median start time. Based on this data, configurations 1, 4, and 6 tend towards faster start times. Configurations 5 and 8, on the other hand, stand out with remarkably high start times and median start times of well over 10 seconds. These results are also confirmed when taking a different look at the data. For each image, we analyze on which host configuration the container start completed fastest and slowest. Considering the median start time, 512 images start fastest on configuration 1 and 367 images on configuration 6. Configurations 5 and 3 stand out for the slowest start times. 439 images start slowest on configuration 5, another 233 images on configuration 3.

In addition to the absolute values of the start time,

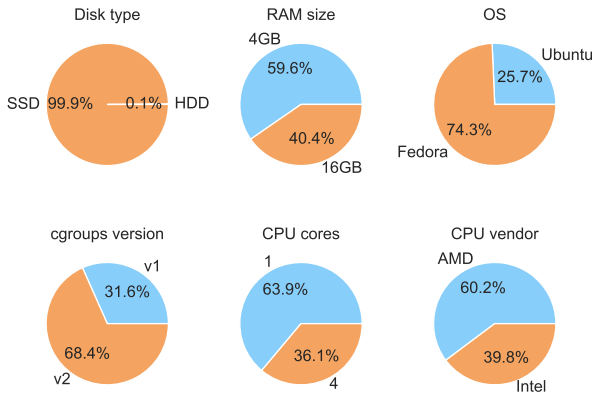

Figure 1: Analysis of which platform parameter setting results in faster start times.

we also look at the variability of the start time measurements for individual images. As mentioned above, we measure a total of 1008 images 20 times on each host configuration. This results in 1008 coefficients of variations for each host configuration. The results are similar to the values for the start time measurements in Table 3. We found that the configurations with faster start times also have a low variability. For example, configuration 1 has an average CoV of 0.087, while the mean CoV of configuration 8 is about 4.5x higher (0.393). Overall, the results shown above confirm our assumption that there are significant differences in the start times for different host configurations. In the following, we will examine in more detail which platform parameters influence start times most.

First, we look which setting (*low* or *high*) was faster for every platform feature and for each of the 1008 images tested. We distinguish the configurations that have different settings for one platform parameter. For example, we select configurations 2, 3, 5, and 8 for the *low* setting of the disk type parameter. This way, we can compare median start times for 1,008 images for *low* and *high* values of every parameter. Figure 1 shows for each platform feature how many images started faster for which feature value. Dominant features tend to have a clear split, while unimportant features tend to be near a 50:50 split. All but one image has lower median start times when we use SSD disks rather than standard HDD disks. Likewise, Fedora tends to have lower start times than Ubuntu, and `cgroups` v2 tends to be faster than v1. The amount of RAM and CPU vendor features seem to be not that important as they are closer to a 50:50 split.

Overall, we see that disk type seems to be the most important platform parameter for start times. Hence, we can confirm our presumption from our previous work. The disk-intensive workload of a container start is the unpacking and provisioning of the container's root file system. Increasing disk throughput should, therefore, be the first consideration when trying the lower start times using host configuration changes. We opted to analyze only the features we can easily set in GCP. We did not analyze the influence of individual software dependencies beyond `cgroups`. Instead, we used the operating system parameter, which acts as a cover for different software dependencies (e.g., different Linux kernel versions).

Finally, we consider platform and image configuration parameters together. To do this, we add the image configuration parameters to the measurements and train a random forest model to predict the start time. As in our previous work, the size of the image is the most important factor for the prediction (39.7%). The second most important feature is the disk type (20.1%). Both features are the dominant ones in our dataset and together determine about 60% of the total importance. Our statement that both platform and image configuration parameters are essential impact factors for start times is confirmed. In this dataset, platform parameters account for 43.8% of the feature importance and image configuration parameters for 56.2%. Hence, both types of parameters are essential for an accurate assessment of container start times.

5 Summary

In our previous study on container start times, we saw that start times are impacted by the host that executes the container. This paper looked deeper into the influence of selected platform parameters. While a representative study as for the image configuration parameters was not possible, we focused on selected parameters regarding the host CPU, RAM, OS, and disk. We found that the disk type is the most impactful parameter, with SSD disks offering significantly higher throughput than HDD disks. However, when combining image configuration and platform parameters, we saw that both feature classes must be considered to assess container start times accurately.

Acknowledgements

This work was funded by the Deutsche Forschungsgemeinschaft (DFG) – 510552229.

References

[1] R. L. Plackett and J. P. Burman. "The design of optimum multifactorial experiments". In: *Biometrika* (1946).

[2] M. Straesser et al. "A Systematic Approach for Benchmarking of Container Orchestration Frameworks". In: *ICPE*. 2023.

[3] M. Straesser et al. "An Empirical Study of Container Image Configurations and Their Impact on Start Times". In: *CCGrid*. 2023.

[4] A. Ebrahimi et al. "Cold start latency mitigation mechanisms in serverless computing: taxonomy, review, and future directions". In: *JSA* (2024).