

BEST PRACTICES FOR VIRTUOZZO CONTAINERS FOR LINUX

Using I/O Limits to Maximize Container Performance

Q2 2013

Table of Contents

Introduction.....	3
Test of I/O Limit Range.....	3
Tests of Lower Limits.....	4
Tests of Upper Limits.....	4
Bandwidth Limits for Backup, Restore, and Migration Processes.....	6
How to Set I/O Limits.....	6
Using the Command Line Interface.....	6
Using the PVA Interface.....	7
Using the POA Interface.....	7
Conclusion.....	8

Introduction

I/O limits were introduced in Parallels Virtuozzo Containers for Linux 4.6 with Kernel 2.6.18, and in Parallels Server Bare Metal 5 with Kernel 2.6.32 and is now available in Parallels Cloud Server 6.0. Used in conjunction with CPU scheduling, these I/O limits provide very granular and powerful controls that help you avoid performance degradation in situations where high disk I/O activities in one container could slow down the performance of other containers on the server – situations that are sometimes referred to as “container penetration.”

There is no one best I/O limit setting to use: typically, settings will differ from one server to the next because the Input/Output Operations per Second (I/OPS) throughput of disk subsystems themselves vary heavily, depending on the disk performance, RAID levels, and type of storage. Network bandwidth limits are another important factor to consider when setting the upper and lower I/O limits.

This document will guide you in the best practices for setting I/O limits for any given server, taking all relevant factors into account. It covers only DAS (Direct Attached Storage) and does not describe settings for SAN-like storage solutions (e.g. Dell EqualLogic), as these solutions usually have separate mechanisms for avoiding I/O bottlenecks.

Test of I/O Limit Ranges

I/O limit parameters include a lower limit (minimum I/O bandwidth required to avoid performance degradation) and an upper limit (maximum I/O bandwidth that will prevent container penetration from degrading performance). In order to establish the useful range of the upper and lower limits, we performed testing on an example server.

The system we used for measurement was a Dell PowerEdge 2950 with a 1 QuadCore Xeon 5110 CPU with 1.6 GHz, 10GB of RAM, and 4 Maxtor ATLAS 10K5 73GB SAS drives in a RAID5 configuration. The system had 40 containers, each running 64-bit CentOS 6 and Parallels Plesk Panel 10.

Tests of Lower Limits

The lower limit test restarted all containers, booting up four containers at a time in parallel. The disk speed was gradually increased in 250-byte steps until reaching 2,500 bytes per second; then in 2,500-byte steps until reaching 10,000 bytes per second; and then in 25,000-byte steps until reaching 100,000 bytes per second.

Figure 1 shows the point at which container reboot time begins to grow significantly. In the figure, the lowest possible useful I/O throughput value – that is, the lowest possible value that won't negatively affect reboot time – is 10,000B/sec. In practice, however, the lower I/O limit should be much higher than that, to allow additional bandwidth for handling short periods of intensive I/O. In a network with a 100Mbit or greater bandwidth, the best practice is to set the lower limit at 10MB/sec. This provides enough bandwidth for common Web applications, including Parallels Plesk Panel.

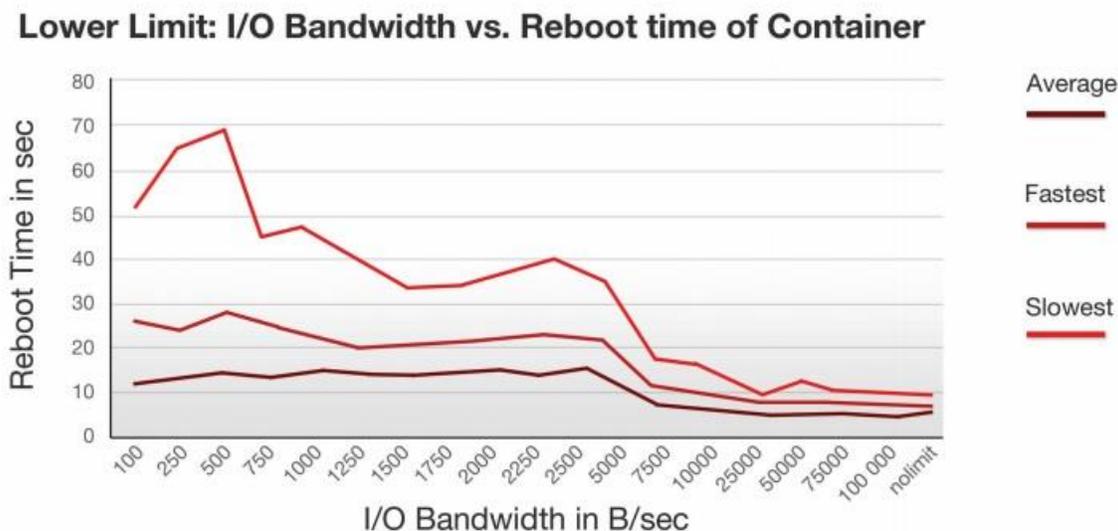


Figure 1. Effects of different lower I/O limits on container reboot time.

Tests of Upper Limits

The upper limit test was performed similarly to the lower limit test, but it included a 41st container that created heavy disk bandwidth utilization. We measured the maximum available I/O bandwidth by running the dd utility several times and writing a file twice the memory size to

the /vz partition. We started with an upper limit of 10 MB/sec and increased the value in 1MB steps.

In this test, the maximum disk bandwidth that could be used by the write operation was 28MB/sec, and reboot times increased dramatically as the upper limit approached that 28MB/sec limit, with 21MB/ sec being the highest upper I/O limit that did not increase reboot time. While your own maximum usable upper limit – that is, the maximum value that won't increase reboot time – will vary according to your specific hardware and drive configurations, it's always best to not allow any container to consume 100% of available disk bandwidth. That's because the physical server will need some bandwidth to read and write data to stabilize the workload. In addition, concurrent I/O will further decrease the maximum throughput. For these reasons, the best practice is to set the upper limit at 75% of the maximum disk bandwidth. Even if your system can deliver more I/O bandwidth, the 75% value will typically be enough to support all applications and end-customers while at the same time protecting the system from container penetration – especially since most customers will be comparing performance with that of a traditional SATA drive.

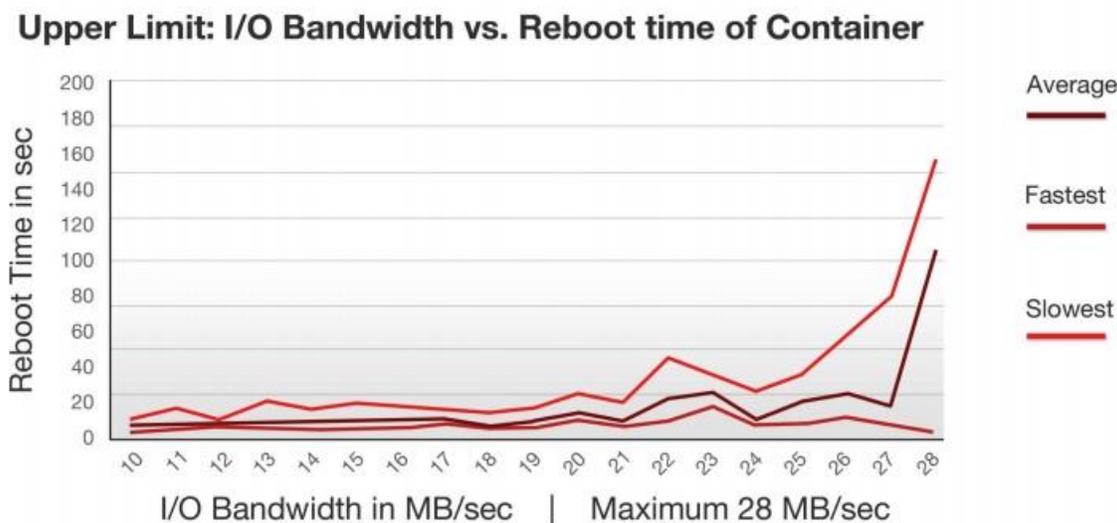


Figure 2. Effects of different upper I/O limits on container reboot time.

Bandwidth Limits for Backup, Restore, and Migration Processes

In addition to setting the lower disk I/O limit at 10MB/sec and the upper limit at 75% of the maximum disk bandwidth, it's a good practice to also set an upper limit for the bandwidth that can be used by backup, restore, and migration processes. That way, you'll avoid performance bottlenecks during these operations. As a best practice, we recommend using the same upper limit as you do for containers: 75% of the maximum disk bandwidth. To set this limit, use the `/etc/sysconfig/vz` file parameters, as shown in Table 3.

Table 1: Limiting Bandwidth for Backup, Restore, and Migration Processes

Parameter	Setting
VZ_TOOLS_BCID	Specify the container ID used to limit backup, restore, and migrate operations. The container ID must be free and should never be assigned to containers.
VZ_TOOLS_IOLIMIT	Specify the limit in bytes per second for disk I/O operations

How to Set I/O Limits

You can set I/O limits in any of three ways: through the Command Line Interface (CLI), through the Parallels Virtual Automation (PVA) interface, or through the Parallels Operations Automation (POA) interface.

Using the Command Line Interface

To set I/O limits via CLI, use the following command:

```
$> vzctl set --iolimit --save
```

Where:

Ctid = The container for which you are setting the I/O limits.

Bandwidth = I/O limit assigned to container. If no suffix is provided, the value is assumed to

be in megabytes per second. Available suffixes are:

- b, B -- bytes
- k, K -- kilobytes
- m, M -- megabytes
- g, G -- gigabytes

Using the PVA Interface

Figure 3 shows how you can use PVA to configure a container. You first enable the I/O limit by checking the appropriate “Limited” box and then enter the appropriate value.

Disk Quota Parameters	
diskinodes	200000 : 220000 inodes Number of total disk inodes. Essentially this is a total number of files and directories which can be created. Internet News Server can be an example of a server which create considerably higher number of files.
diskspace	1048576 : 1153434 1K-blocks Total size of disk space that can be consumed by the Virtual Private Server in 1-Kb blocks. When specified by the 'quotatime' parameter.
iolimit	0 MBps The disk I/O rate that can be consumed by the Container, in MByte/s
quotatime	0 seconds

Figure 3. Setting a containers limits using POA.

Using the POA Interface

Figure 4 shows how to use POA to configure the IO Limits for a container. Please navigate to Top >> Service Director >> Virtuozzo Manager >> VPSs to access this control panel

▼ Disk Quota						
Parameter	Limited	Current Usage	Soft Limit	Hard Limit	Units	Description
diskspace	<input checked="" type="checkbox"/>	n/a	1024	1126	MB	Total size of disk space that may be consumed by the Container
diskinodes	<input checked="" type="checkbox"/>	n/a	200000	220000	inodes	Number of disk inodes
quotauidlimit	<input type="checkbox"/>	n/a	n/a	Disabled	pcs	Number of user/group IDs
quotatime	<input checked="" type="checkbox"/>	n/a	n/a	0	seconds	Grace period of Container disk quota
ioprio	<input checked="" type="checkbox"/>	n/a	n/a	4	units	Container priority for disk I/O operations
iolimit	<input type="checkbox"/>	n/a	n/a	Unlimited	Bps	Disk I/O rate limit
iopslimit	<input type="checkbox"/>	n/a	n/a	Unlimited	nps	Input/Output Operations Per Second

Figure 4. Setting a container's I/O limits using PVA

Conclusion

Setting the proper I/O limits on containers will help you prevent high disk I/O activities in one container from degrading the performance of other containers on the same server. In general, a good lower limit is 10MB/sec, which provides enough bandwidth for common Web applications, including Parallels Plesk Panel. A good upper limit is 75% of the overall disk throughput, as it reserves enough capacity to enable the physical server to rapidly handle operations like migration, backup, and reinstallation, even if a container should penetrate the disk subsystem. You should also set the bandwidth available to backup, restore, and migration processes to 75% of the overall disk throughput. By setting the appropriate I/O limits in each case, you will maximize the performance of all containers on the server and prevent performance degradation.