

# Using Oracle's vdbench tool to create a ~5.5:1 reducible dataset on a Pure Storage FlashArray

Creating a standard dataset that is comprised of data that is both compressible and dedupable when conducting storage performance tests on a data-reduction storage array. In the "Using Oracle's vdbench tool to test the data-reducing capabilities of your storage array" blog we showed 3 simple examples that one could use to create standard datasets to test the data-reducing capabilities of a storage array.

We always discourage customers from using synthetic storage performance tests if possible. Instead we encourage them to conduct storage performance tests using their unique set of applications. This is the best way to find out how a data-reduction storage array will actually perform with their particular datasets, concurrency levels, workloads, and workflows. To make this more practical for customers, Pure Storage has the "Love Your Storage™ Guarantee."

However, we realize that often using synthetic load-generation tools is the only practical solution. Thus, in order to make synthetic storage performance tests more realistic and useful for storage performance sizings, we encourage the creation of datasets that reflect the average reducibility rate that our customers are able to achieve on their datasets. The Pure Storage Flash Reduce Ticker is updated frequently to reflect the average data reduction that customers' FlashArrays report to Pure1. Usually, it hovers between 5.0:1 and 5.5:1.

In this blog we will share the scripts for Oracle's vdbench tool that can be used to create a ~5.5:1 dataset on a Pure Storage FlashArray that is comprised of both compressible and dedupable data. Of course, every storage array has different data-reduction technologies, so this same dataset may reduce very differently on different storage arrays. But, we know that this is the average reducibility rate that Pure Storage customers get on their FlashArrays, so we think that this is a very defensible dataset to test with.

To create this dataset, we use a combination of "Oracle vdbench 5.04.03 4/4k/5" data and "Oracle vdbench 5.04.03 3/4k/4" data.

"Oracle vdbench 5.04.03 4/4k/5" data is generated using the following Oracle vdbench 5.04.03 parameters:

```
dedupratio=4
dedupunit=4k
compratio=5
```

"Oracle vdbench 5.04.03 3/4k/4" data is generated using the following Oracle vdbench 5.04.03 parameters:

```
dedupratio=3
dedupunit=4k
```

compratio=4

We've found through experimentation that if you use workloads comprised of 45% "Oracle vdbench 5.04.03 4/4k/5" data, and 55% "Oracle vdbench 5.04.03 3/4k/4" data, it will produce a dataset that reduces to ~5.5:1 on a [Pure Storage FlashArray](#). Of course, over time, just like in a real-world Pure Storage customer environment, as the deep-reduction engines of the FlashReduce feature of the Purity Operating Environment get more time to work on the dataset, it might reduce even more. However, under performance testing pressure, the reducibility of this dataset does stay pretty close to 5.5:1.

In our opinion, this is a good dataset because it is entirely comprised of data that is both compressible and dedupable, rather than having relatively large slices of the dataset that are non-reducible-only or compressible-only.

We still break up the dataset into 20 slices by default, however, you could have many more slices. The only reason we have slices at all is because, at this time, you cannot create a single Oracle vdbench script that uses more than one definition for dedupratio, dedupunit, compratio.

The 20 "Oracle vdbench 4/4k/5" sub-slices are filled with Oracle-vdbench-generated data from scripts like this:

```
dedupratio=4
dedupunit=4k
compratio=5
messagescan=no
include=/root/AFA/pure_fill_5-5/host_definition.vdb
include=/root/AFA/pure_fill_5-5/storage_definition.vdb
wd=default
wd=fill,rdpct=0,seekpct=0,xfersize=(4k,12,8k,17,16k,21,32k,30,64k,7,128k,7,256k,6),range=(0.00g,7
3.47g),sd=sd*
rd=default
rd=fill1,wd=fill,iorate=max,interval=60,elapsed=172800,maxdata=2351.04g
```

The 20 "Oracle vdbench 3/4k/4" sub-slices are filled with Oracle-vdbench-generated data from scripts like this:

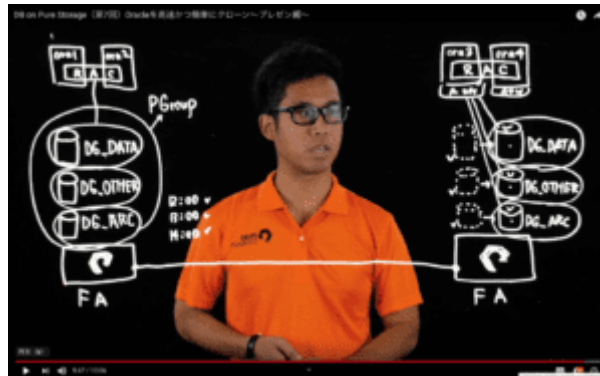
```
dedupratio=3
dedupunit=4k
compratio=4
messagescan=no
include=/root/AFA/pure_fill_5-5/host_definition.vdb
include=/root/AFA/pure_fill_5-5/storage_definition.vdb
wd=default
wd=fill,rdpct=0,seekpct=0,xfersize=(4k,12,8k,17,16k,21,32k,30,64k,7,128k,7,256k,6),range=(73.47g,
163.27g),sd=sd*
rd=default
rd=fill1,wd=fill,iorate=max,interval=60,elapsed=172800,maxdata=2873.60g
```

Scripts to create this dataset of various sizes can be found [HERE](#).

The scripts are designed to be used on our standard testing configuration of 1 "command" VM and 8

“worker” VMs. These scripts were tested using [CentOS 7](#) VMs, but can be modified to work on other operating systems that vdbench supports.

[Oracle’s vdbench tool \(5.04.03\)](#) was used in these tests.



動画「Oracle を高速かつ簡単にクローン ~ プレゼン編 ~」

A script named “purevol\_create\_32\_vdb\_luns” can be used at the Purity Operating Environment’s command-line interface to create the 32 Purity volumes that hold the dataset.

On Linux, simply unzip these scripts into the “/root/AFA/pure\_fill\_5-5” subdirectory of the “command” VM. To execute these scripts, invoke the “./do\_pure\_fill\_all” bash script.

We invite you to use these scripts to create a dataset on your current data-reduction storage array. If you get less than around 5.5:1 [data reduction](#), and having increased performance with consistently lower response times, while at the same time providing higher levels of data reduction is important to you, please give us a call.

Finally, if you want to be involved in a helping to improve vendor-neutral performance testing of Solid State Storage Systems, please consider joining the [SNIA S4TWG](#). This technical working group currently has the participation of many storage performance testing industry experts, and we would welcome you joining us.