

Calculating Block Size Using the Powershell SDK



The Pure Storage PowerShell SDK covers all aspects of management and automation for the FlashArray FA-400 series and the FlashArray//m series. In many of the discussions with our customers, they are most interested in reporting. On several occasions customers have asked how can I get the specific block size to create their own custom graph reports. In this blog article I'll provide the details on how to calculate the block size by creating a custom cmdlet that leverages the PowerShell SDK.

Below are the results from a 1 hour workload test using the below command. The test below test sets a defined read block size of 32,768 and a write block size of 16,384. I did this to ensure that the calculations can be validated using known block sizes to verify against what we are showing in the Pure Storage web management interface.

```
nohup pureload -i pct=.6,op=read,size=32768 -i pct=.4,op=write,size=16384,dtype=h20 -m maxoff=1
,initthreads=10,maxthreads=10,interval=86400,warmup=1,status=1,maxlat=20000 /dev/pslun-Barkz-
PURELOAD
```

The screenshot below shows a 1 hour view of the above workload. To fully understand the calculation we will pinpoint a specific time 14:46:02pm. At this specific time Read = 30.28K, Write = 20.14K and Average R/W IO Size = 25.61KB.

The management interface does all of the calculations both in the Dashboard and Analysis tabs for quick review. But what if you want to pull the live data yourself? I thought you'd never ask!

In the Pure Storage PowerShell SDK there is a cmdlet, Get-PfaVolumeIOMetrics that can be used to pull data from the FlashArray based on a time range. The example code below retrieves 1 hours worth of I/O data from the volume named Barkz-PURELOAD.

```
$Creds = Get-Credential
$FlashArray = New-PfaArray -EndPoint 'csg-fa420-2' -Credentials $Creds -IgnoreCertificateError
Get-PfaVolumeIOMetrics -Array $FlashArray -VolumeName 'Barkz-PURELOAD' -TimeRange 1h |
Select-Object reads_per_sec, writes_per_sec, output_per_sec, input_per_sec, time
|
Format-Table -AutoSize
```

Below are the results from the code above. I am only displaying 5 results. You can see from the results we provide information every 30 seconds.

```
reads_per_sec writes_per_sec output_per_sec input_per_sec time
-----
34633      23063    1134839945   377860915  2015-08-21T21:45:02Z
31068      20734    1018022025   339712956  2015-08-21T21:45:32Z
```

31238	20803	1023590400	340836352	2015-08-21T21:46:02Z
31343	20901	1027057254	342446353	2015-08-21T21:46:32Z
34290	22803	1123602705	373610906	2015-08-21T21:47:02Z

The data above is the same data that we use to calculate what is displayed in the management interface. The basic calculation that our web management interfaces uses is the following:

$$((\text{output_per_sec} + \text{input_per_sec}) / (\text{reads_per_sec} + \text{writes_per_sec})) / 1024$$

Using the time we pinpointed, 14:46:02pm, we can calculate the block size.

$$((1023590400 + 340836352) / (31238 + 20803)) / 1024 = 25.60\text{KB}$$

The calculation above uses 1024 as the division because we know the IO size in this case is in Kilobytes. The management interface dynamically calculates the correct unit based on the raw number. For example is an IO size happens to be in the Megabytes then it would use "1024*1024"; the same applies to GB "1024*1024*1024". Keep this in mind as you create custom calculations in Microsoft Excel or other tools.

More on PowerShell

The above shows how to calculate that one specific data point, now let's retrieve the full hours worth of data and export as a CSV to be used with Microsoft Excel 2013. Building on the sample code from above, we add Export-Csv to the pipeline and can export as a CSV.

```
$Creds = Get-Credential
$FlashArray = New-PfaArray -EndPoint 'csg-fa420-2' -Credentials $Creds -IgnoreCertificateError
Get-PfaVolumeIOMetrics -Array $FlashArray -VolumeName 'Barkz-PURELOAD' -TimeRange 1h | `
Export-Csv -Path 'C:TEMPPureStorage-BlockSize-1HR.csv' -NoTypeInformation
```

Opening up the PureStorage-BlockSize-1HR.csv file with Microsoft Excel will show all of the data collection points for the past hour. With a few quick steps in Excel you can create a table and add the basic calculation.

Select cell A1, CTRL+T will automatically create a table. In column "I" add Block Size and that will automatically extend the table to include the newly added column. With a table created the calculation can be created using the column names as follows:

Once the calculation has been created in cell "I2" it will automatically be added to all subsequent cells. Now once that is completed you can get as creative as you want with graphs. For example below is a simple graph that shows the block size calculations for the 1 hour we exported above.

Try out [FlashBlade](#)

Happy graphing!
barkz