

Characterizing WorkloadWisdom Data Center Workloads

Virtual Instruments Best Practices

May 21, 2018

Version 7.0

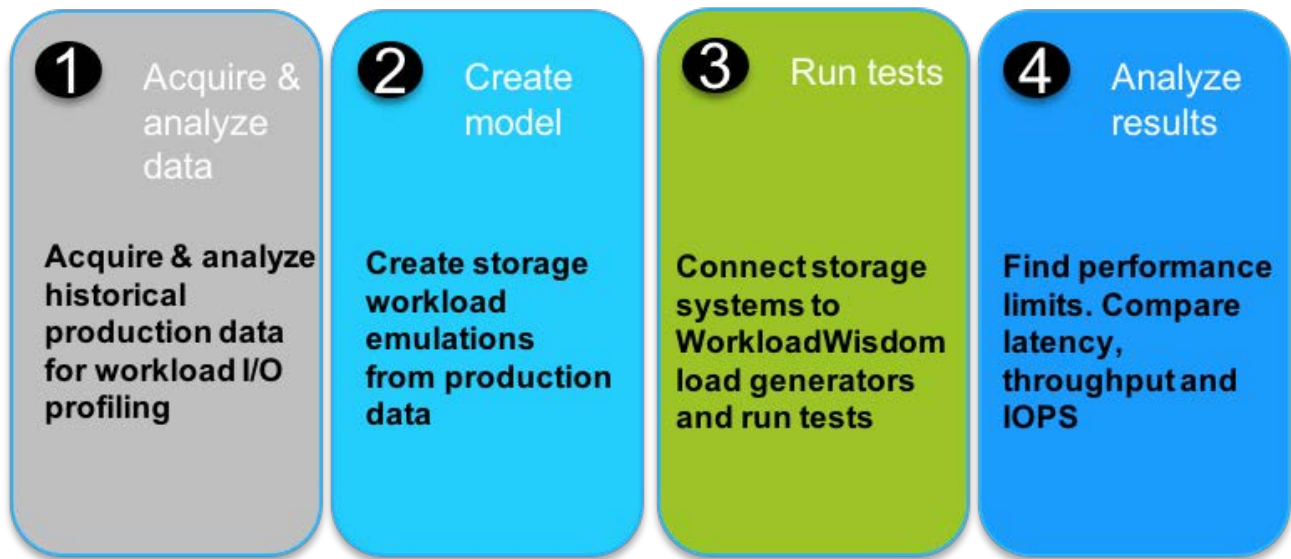
TABLE OF CONTENTS

Abstract.....	3
Virtual Instruments Services	3
What to Characterize	3
The Importance of Granularity in Workload Characterization.....	4
Workload Details	4
Impact Characterization.....	7
Granularity of Characterization.....	8
Elements for which the metrics are collected	10
How to characterize workloads.....	12
Workloads.....	13
Composite Workloads.....	14
Importing a Workload Data File.....	14
Workload Data Importer using data from the VW hardware probes	19
Workload Profiles using data from the VW ProbeVM (for vSphere).....	20
Storage Infrastructure Vendor Examples.....	20
EMC VNX UNISPHERE®.....	21
EMC VMAX UNISPHERE®	23
EMC Isilon.....	24
HDS.....	25
HPE 3PAR.....	27
HPE EVA	28
IBM® XIV	29
NetApp® FAS	32
Pure Storage®	33

Abstract

Virtual Instruments combines a storage workload modeling application – WorkloadWisdom, with purpose-built load generation appliances and data capture probes, to help storage architects and engineers to accurately characterize storage performance. Step one is to acquire and analyze product data to characterize the workload model. In 2015, Virtual Instruments changed the industry by automating the analysis of production workload data, via the Workload Data Importer feature of WorkloadWisdom.

This document provides a summary and details for characterizing network storage workloads and determining the level of data required for a good, better, or best outcome. The more complete the data reported by an array or performance monitor, and the higher the resolution in terms of reporting period, the more closely a production application can be accurately modeled for testing. We describe the details of what should be monitored and provided to produce a superior workload model from a production workload.



The goals of this document are two-fold:

1. Outline ways to characterize workloads for Enterprise data center applications. The goal is intended to cover the most common characteristics that are readily available information and impact developing a good emulation of the workload. It is not intended to cover corner case characteristics that would be required in order to test the full functionality of a storage array. The intent is to enable the development of a realistic enough workload to be able to compare different devices, configurations, firmware versions, or to detect degraded components in an infrastructure. Characterizing a workload that would be complete enough to do a full storage manufacturer regression test of a storage subsystem is outside the scope of these recommendations.
2. Describe how the Workload Data Importer simplifies and automates the analysis of production storage workloads. While understanding the section “What to Characterize” is useful, it’s no longer necessary, as the Importer now does the heavy lifting.

Virtual Instruments Services

As the leader in infrastructure performance optimization, the Virtual Instruments Professional Services organization helps teams to characterize their workload and to model configurable workloads to test ‘what if’ scenarios against their most common workloads.

What to Characterize

There are 4 basic areas to consider when characterizing a workload for a storage environment.

1. Description of the size, scope and configuration of the environment, including number of servers, LUNs, volumes, and shares
2. The patterns that describe when, how frequently and in what ways data are accessed
3. IOPS and throughput rates during the time-period data is gathered

- The impact on the network subsystem in addition to the patterns observed on the array itself. Though this information isn't used directly to model the workload, it enables an impact comparison to be made between the emulated traffic and the actual production traffic. This helps measure how representative the emulated traffic is to real-world traffic

The Importance of Granularity in Workload Characterization

Besides knowing what to characterize, it is important to understand the granularity of the collected data. Data granularity determines the quality of the resulting emulated workloads in terms of a good, better, or best characterization. It is important to consider:

- How often and how long the data is collected
- How completely the data represents workload elements like LUNs, volumes/mount points, shares, directories/files and LBAs.
- How detailed the access patterns are in terms of data and metadata protocol command coverage.
- How detailed the information is in terms of access patterns and request sizes.

High Level Guidelines

The realism of the Workload Model that can be created from the analyzed Workload Data depends heavily on the fidelity of the workload data that is provided by the storage arrays and performance monitoring tools. As there is not an industry standard for the data structure provided by various storage arrays and performance monitoring tools, and as there are different ways to arrive at the same metrics even though the data sources are different, only a high-level guideline can be provided here as a starting point, and that a deeper technical discussion must be held to define the exact data set that is needed to create a parser / tool that simplifies workload modeling.

Protocol / General	Description	Model Quality ¹
FC / iSCSI	Per-LUN R/W, Block Size distribution, KPIs ²	Best
FC / iSCSI	Per-LUN R/W, KPIs	Good
FC / iSCSI	Per-array R/W, KPIs	Minimum
NFS / SMB	Per-volume R/W, Metadata buckets, block sizes, KPIs	Best
NFS / SMB	Per-array commands AND per-volume R/W/Other, KPIs	Good
NFS / SMB	Per-array commands OR per-volume R/W/Other, KPIs	Minimum
General	Data interval: 1s – 1min	Best
General	Data interval: 1min – 5min	Good
General	Data interval: 5min – 10min	Minimum

Most commercial storage arrays and performance monitoring tools provide data sources that fall under the Good category. The VirtualWisdom solution from Virtual Instruments and a few commercial solutions provide data sources that fall under the Best category above.

Workload Details

What to Characterize

Storage environment

Understanding the storage environment differs for file, block and object storage. Each environment has unique characteristics that must be understood to create and map an emulated workload similar to the observed production environment.

File (NAS)

¹ The Model Quality is assessed only by the Virtual Instruments Product Management team, in the context of creating realistic Workload Models that sufficiently represent the observed Production Workload Data

² Key Performance Indicators are Throughput, IOPS, Latency

For file, the following environmental characteristics should be determined:

- The number of clients and servers/clusters
- The number of clients per server/cluster
- The authentication medium for clients to gain permission to access the servers/clusters
- Shares and mount points
- File size distribution
- Sub directory and tree depth distribution
- Average number of files per directory and server/cluster

Block (SAN)

- For block, the following environmental characteristics should be determined:
- The number of physical initiators in the environment (HBA or NIC's)
- Number of storage ports
- Port bandwidth
- The average number of virtual ports per physical port
- Number of Logical Units (LU's) per storage port
- Size of each LUN
- IO hotspots of each LUN
- Zoning configuration
- The initiator to target fan-in ratio (the average number of initiators that access each target port)
- For Fibre Channel, the queue depth setting in the switch environment for the server HBAs

Object

- Number of clients and servers
- Average and maximum open connections
- Authentication medium for data access
- Object size range distribution
- Number of buckets or containers
- Number of objects per bucket/container

Access patterns

The access patterns are key to understanding how frequently and by what means storage is accessed. This is the area where it is most important to consider varying business cycle access patterns such as average and peak access patterns, the impact of special or periodic business processes, and events. The proper characterization of access patterns is different or file, block, and object environments.

File (NAS)

For file access, it is important to consider the breakdown by percentage of common commands. Often there are far more metadata commands in an environment than read/write commands. The most important commands to analyze for each protocol version are:

NFSv3

- Read KB/s, Write KB/s
- I/O request size distribution (block sizes)
- I/O request locations [percent or regions of large file access]
 - Data access
 - Hot spots
- Read and Write cache hit percentage
- Typical sizes of I/O active regions
- Data content compressibility (data compressed to reduce size on the array)
- Data content de-duplicability (eliminating duplicate data to reduce size on the array)
- CPU utilization
- READ command percentage
- WRITE command percentage
- LOOKUP command percentage
- GETATTR command percentage
- SETATTR command percentage
- REaddir command percentage
- READLINK command percentage

- REaddirPLUS command percentage
- CREATE command percentage
- REMOVE command percentage
- FSSTAT command percentage
- ACCESS command percentage
- COMMIT command percentage

SMB2

- Read KB/s, Write KB/s
- I/O request size distribution (block sizes)
- I/O request location distribution (where data are accessed within large files)
 - Data access
 - Hot spots
- Read and Write cache hit percentage
- Read latencies
- Write latencies
- Typical sizes of I/O active regions
- Data content compressibility (data compressed to reduce size on the array)
- Data content de-duplicability (eliminating duplicate data to reduce size on the array)
- CPU utilization
- READ command percentage
- WRITE command percentage
- CREATE command percentage
- QUERY_DIR command percentage
- QUERY_INFO command percentage
- SET_INFO command percentage
- FLUSH command percentage
- CLOSE command percentage

Block (SAN; Fiber Channel and iSCSI)

There are far fewer block-related metadata commands relative to file access. Most are used for array maintenance. Block applications are responsible for maintaining their own metadata. If block storage is used to store file or object information, a translation layer must be maintained to describe the data and its characteristics, such as directory and file names, access permissions etc. Access to this data looks to an array like block read or write commands. To characterize block data, it is important to understand basic Write/Read command mix, whether the data is accessed sequentially or randomly, I/O sizes, frequently accesses data ranges (hotspots) and data content. The following parameters are used to describe the block-related access patterns:

- Read / Write ratio
- Read Random / Sequential ratio
- Write Random / Sequential ratio
- Distribution of I/O request sizes (block sizes)
- Distribution of I/O request locations [logical block address regions (LBA)]
- Data access
- Hot spots
- Read and Write cache hit percentage
- Read latencies
- Write latencies
- Metadata latencies
- Typical sizes of I/O active regions
- Data content compressibility (repeating patterns)
- Data content de-duplicability of the store data (duplicate files or file segments)

Object

- GET (Read) KB/s
- PUT (Write) KB/s
- I/O request size distribution (block sizes)
- I/O request location distribution (percentage or regions of large object sizes)
- Data access

- Hot spots
- GET and PUT cache hit percentage
- GET latencies
- PUT latencies
- Typical sizes of I/O active regions
- Data content compressibility (data compressed to reduce size on the array)
- Data content de-duplicability (eliminating duplicate data to reduce size on the array)
- Database transactions (Short PUTs and GETs)
- Database analytics (Long GETs)
- Personal storage and backup (Long PUTs)
- Document access (e.g. SharePoint, long GETs and PUTs, short GETs and PUTs)

Load Patterns

Load patterns help to determine how much storage demand and load fluctuates over time. The remaining piece of the modeling puzzle for generating real-world workloads is understanding how the following characteristics vary over time. The following characteristics are common to all protocols, although they may have slightly different meanings.

- IOPs per NIC or HBA
- IOPs per application type
- READ IOPs
- WRITE IOPs
- Metadata IOPs
- Total throughput (MBs)
- Read throughput (MBs)
- Write throughput (MBs)
- Data content compressibility
- Data content dedupability
- Number of Connections

Each protocol has additional characteristics unique to the protocol that are necessary for comprehensive characterization.

File (NAS)

NFS

- Number of locks
- Metadata throughput (MBs)
- NFSv4.1 number of open connections / sessions

SMB2

- Number of Sessions
- Number of Open Files
- Number of Locks
- Number of Notifications
- Metadata throughput (MBs)

Block (SAN)

- Number of pending/queued exchanges
- Number of reservations

Object

Virtual Instruments is working with SNIA and leading storage architects to build a model that accurately describes the load patterns of object-based stores. This material will appear in future revisions of this document.

Impact Characterization

It is often useful to observe the impact of the load pattern effects over time on the following subsystem components

- Controller or Server CPU utilization(s)
- Storage access by Tier for Tiered storage
- Cache hit ratio

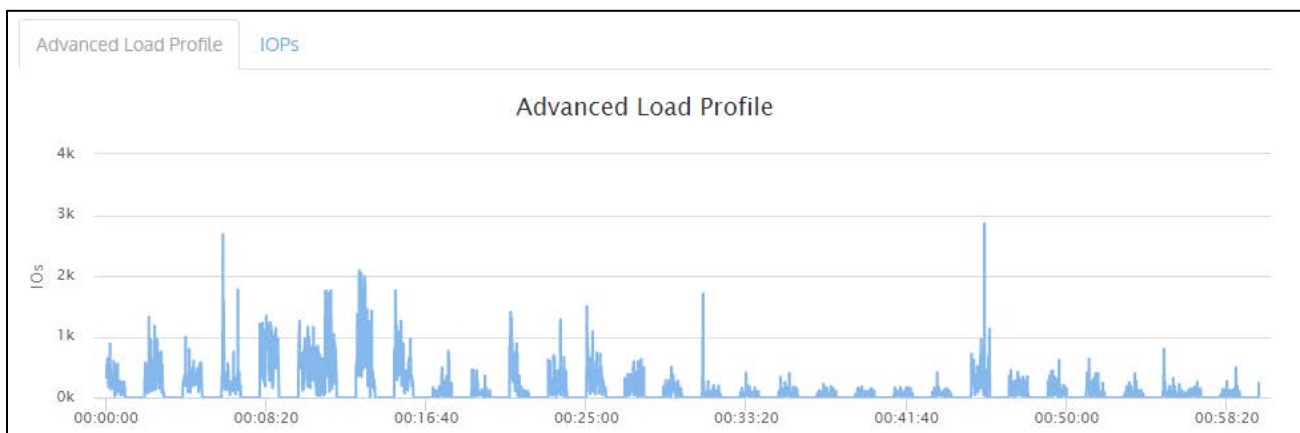
- Raid type weighed hit ratio
- Sequential / Random access percentage (may differ from the perspective of the requestors)
- Latency (ideally broken down by categories such as seek, queue, read, write, etc.)

By comparing these values from the production environment to those in the emulated environment, one can assess how closely the emulation mirrors the production workload. It is best to confirm the workload against the same storage device as used in production before running it against other systems. The workload should be tuned until it closely matches these impacts. Often this can be tuned by adjusting hotspot locations, sizes and access percentages while the command mix and access patterns obtained from the production environment are maintained.

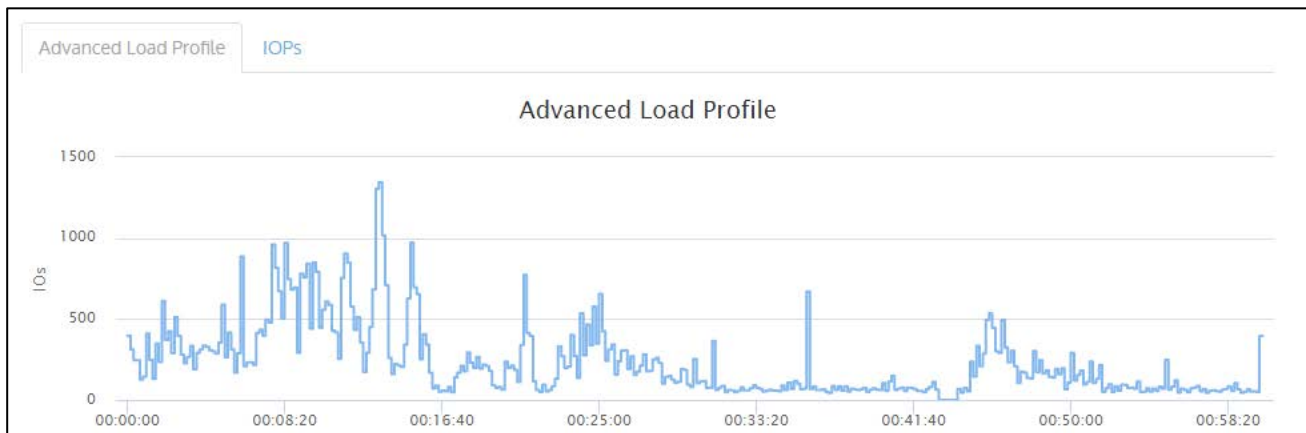
Granularity of Characterization

Time Interval for data collection

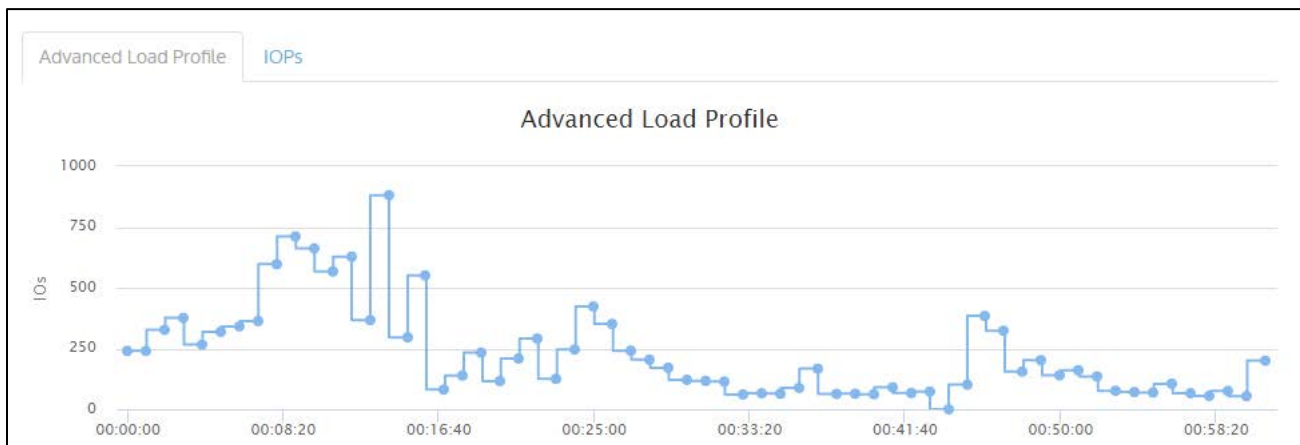
The smaller the time interval the better the workload will match the production environment. Workloads can contain many bursts that will significantly impact performance. It is important to capture these bursts. Here are some examples of the same data provided with different time granularity. From the one second data, traffic bursts of up to 3K per second are clearly displayed:



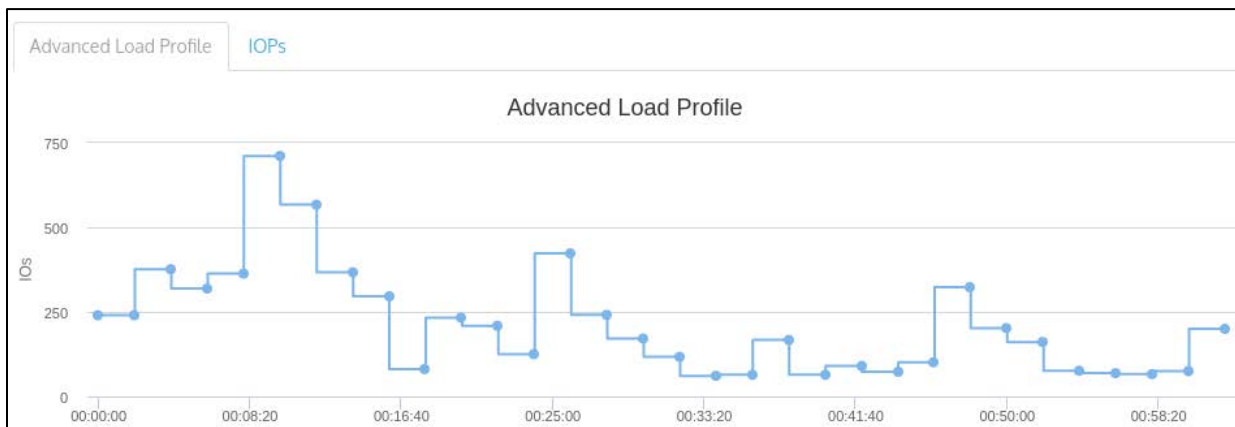
Merely changing the workload from a 1-second to a 10-second interval can cause bursts to be lost. In the graph below, instead of spiking to nearly 3K IOPS, the workload doesn't achieve even 1500 IOPS because one-second data is averaged over the full 10 seconds:



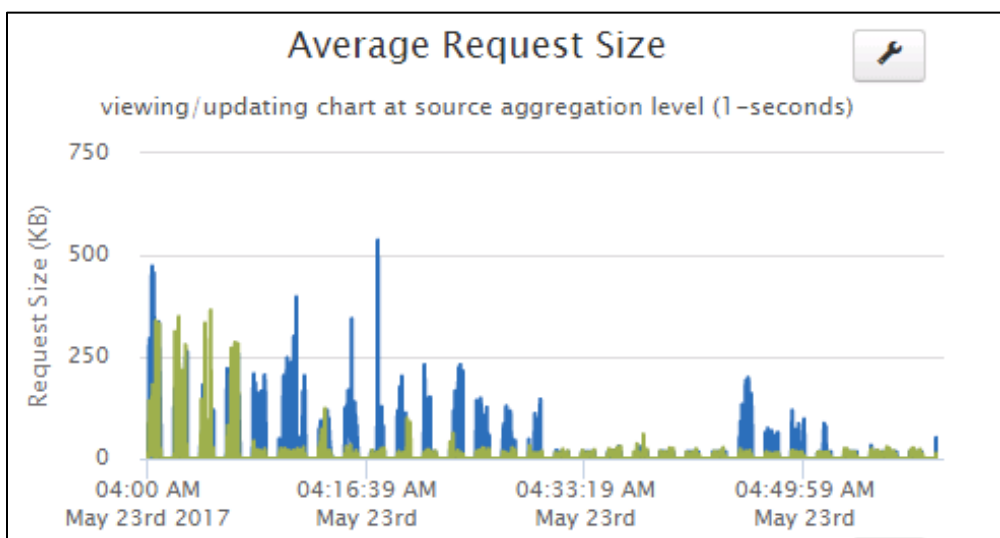
Even worse, at 1-minute granularity the average IOPS rate drops below 1K. Using this period, we will never see the nearly 3K IOPS spikes that were contained in the production workload:



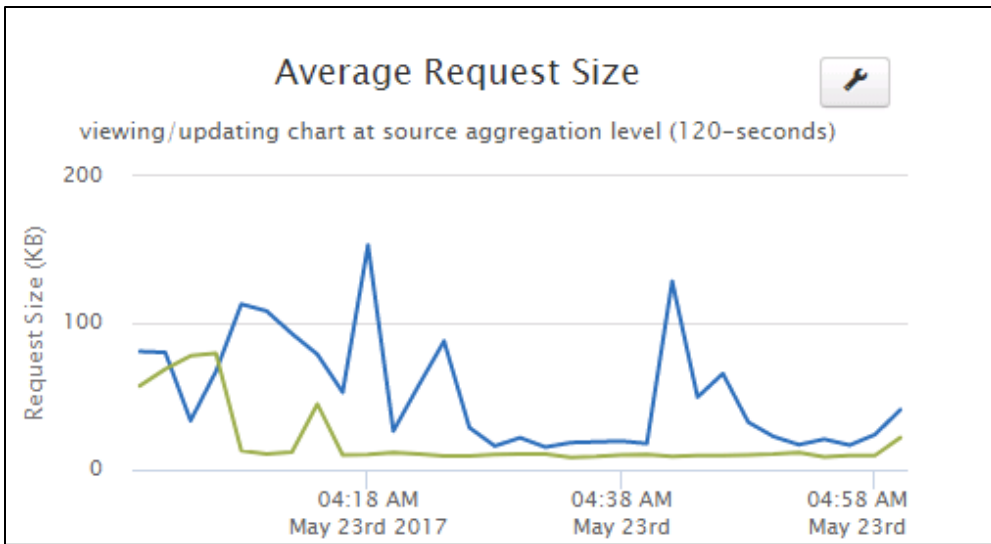
Using a 2-minute interval, IOPS drop below 750, further reducing the accuracy of the workload model:



When granularity is reduced, we also lose information about the request size changes over time. The data at 1 second granularity:



Using a 2-minute reporting period can cause us to miss the large and small requests that are present in the workload:



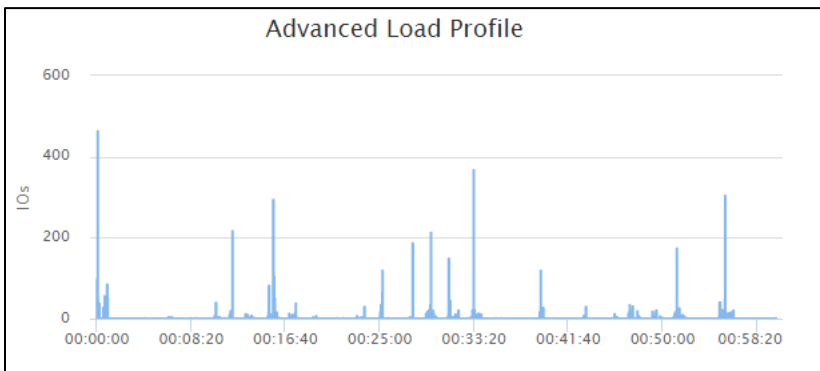
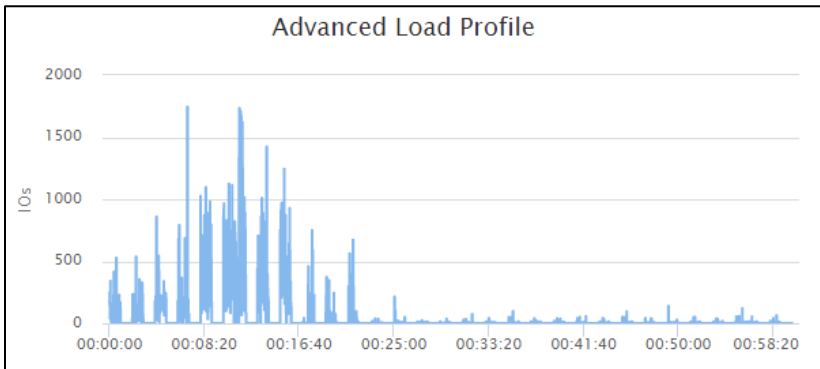
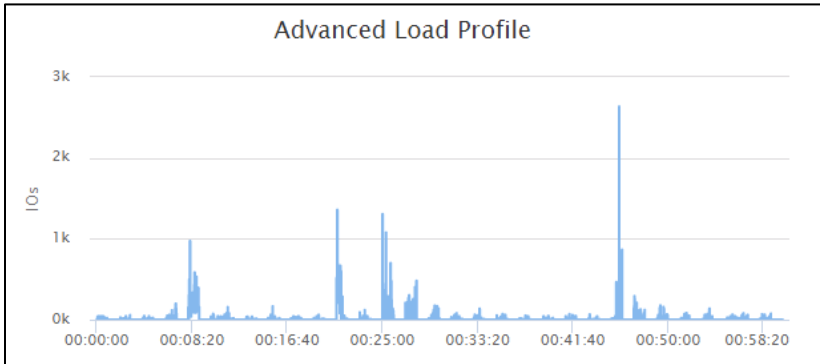
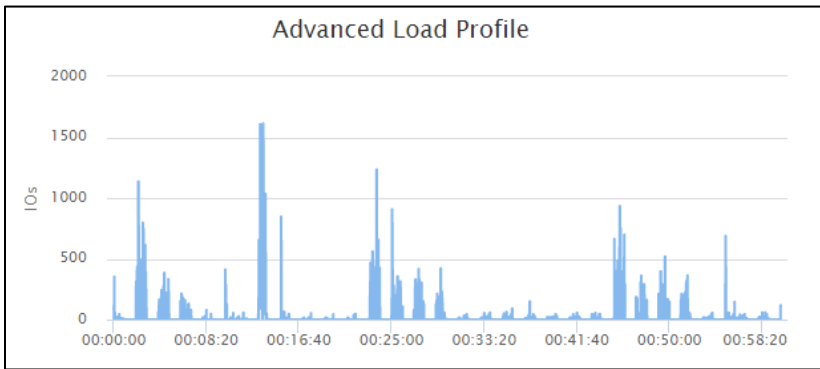
Elements for which the metrics are collected

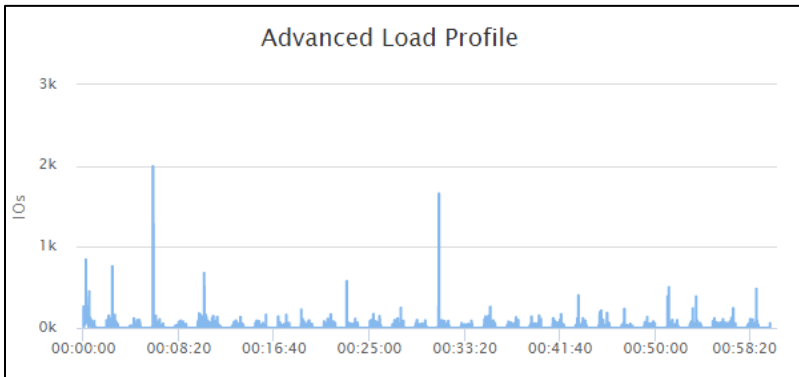
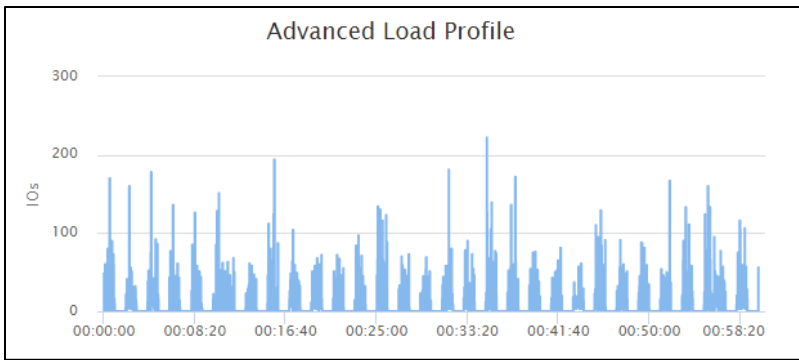
It is also important to break down workload data into per-LUN, Virtual Volume or Mount point, or Share. Often, a significant amount workload traffic is to a few LUNs. In one example, we saw that 2 LUNs had around 25% of the traffic access. From the production workload data, we created 8 workloads, one for the 2 busiest LUNs one each for the remaining 6 LUNs, which received little of the workload traffic. We ran these 8 workloads simultaneously against the array and all metrics indicated the array would keep up with the workload indefinitely with reasonable latency. We then tried creating just a single workload for the environment as we would have done if the data was not available for each LUN. When we ran that workload against the same array configuration, the latency immediately shot up to 130ms, which would be unacceptable for most applications.

For this example, we can see during the time-period some LUNs had almost no activity, while 4 LUNs had the most activity. Also, some LUNs had large block activity and some had small.

Analysis Rule / Workload Patterns	IOPs	Average Latency (ms)	Average Throughput (MBs)	Average Block Size (KB)	Target/LUN Count	Generate IOPs Using
Analysis Rule 1: <input type="checkbox"/> Create a composite workload made up of 8 HiFi FC Workloads using the average IOPs for each workload type. Quality: 97 % <input type="checkbox"/> Check All						
Test_1_Second 53 IOP	53	0.63	5	90	4 i	<input checked="" type="checkbox"/>
Test_1_Second 29 IOP	29	0.09	2	53	4 i	<input checked="" type="checkbox"/>
Test_1_Second 20 IOP	20	0.03	0.22	11	29 i	<input checked="" type="checkbox"/>
Test_1_Second 16 IOP	16	0.1	1	75	2 i	<input checked="" type="checkbox"/>
Test_1_Second 9 IOPs	9	<0.01	0.09	9	28 i	<input checked="" type="checkbox"/>
Test_1_Second 2 IOPs	2	0.02	0.02	13	5 i	<input checked="" type="checkbox"/>
	0.87	0.01	0.02	15	2 i	
	0.38	0	<0.01	3	427 i	

Activity for each of the 6 created workloads:





Details of Access Patterns Command Coverage

For Block protocols, most arrays track read and write activity and in some cases a broad category called ‘other’. Most block workloads are mostly read and write activity, except when certain maintenance tasks are being performed. It is therefore best to focus the characterization on the read and write activity and if any maintenance is being performed, to run similar maintenance activity while the workload is being replayed.

For File protocols, there are often a good deal of meta commands or activity that isn’t related to reading or writing data within a file. In some cases, metadata commands can represent 95% of the workload. This is frequently the case for webserver applications, file servers and applications that deal with lots of small files. For databases and OS files such as VMDKs, there are frequently fewer metadata commands, so it can be less important to accurately replay the metadata commands.

Ideally, arrays would track all the metadata commands for all Virtual Volumes, Shares, or Mount points. Often they merely track Read, Write and a broad category titled “Other” commands. In some cases, they may track a few of the more common metadata commands, while not collecting detailed data for other metadata commands. In those cases, it is important to see if the array at least tracks all data and metadata commands at the array level. In that case, it’s at least possible to use the overall ratio of the “Other” commands to modify the workload model to increase accuracy of the imported workload by combining this data with more detailed data.

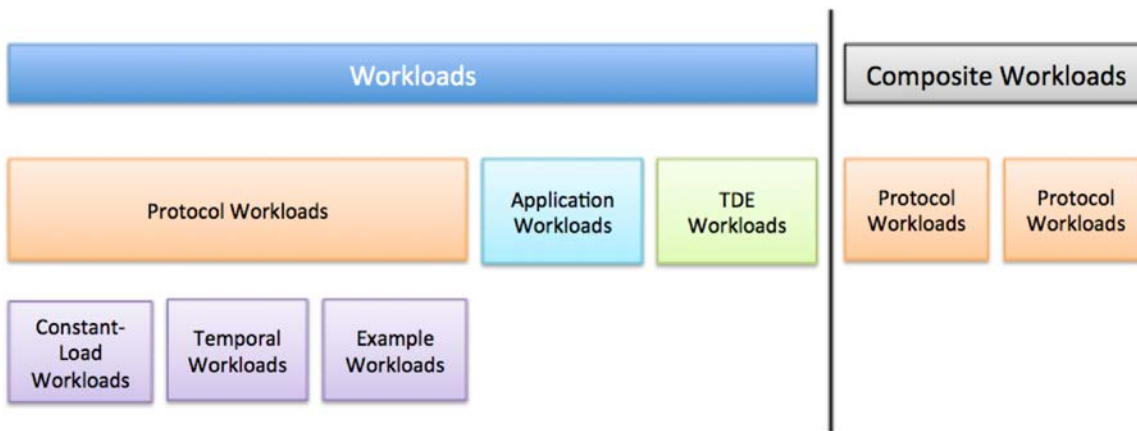
Access Patterns Request Size Details

For all protocols, request sizes (often referred to as block sizes) for reads and writes is a very important characteristic of the workload. Unfortunately, few arrays track complete command sizes or bucket size distributions for Writes and Reads. Ideally, as with access patterns, this should be tracked at the LUN, Virtual Volume, Share or Mount point level. If not, it is still helpful if the array tracks that information at the array level, which can then be applied to the average block size reported for similar workloads.

How to characterize workloads

Types of Workloads

A workload can be broadly defined within a storage context as the storage I/O requests made by application systems against storage infrastructure. When these requests are analyzed in terms of IOPs, read or write percentages, sequential or random behavior, block size distribution their characteristics can be used as the basis for modeling that Workload. The resulting model can be used to emulate the Workload to support numerous activities across the storage lifecycle development, pre-production or production uses like testing, technology or design evaluation, capacity planning, performance analysis and optimization.



Workloads

WorkloadWisdom provides powerful, easy to use, storage Workload modeling and emulation.

- Protocol Workloads: It provides out-of-the-box Protocol Workload models for key file, block and object protocols that make it easy to create and model storage Workloads specific to your environment or applications. A Protocol Workload is not based on any specific application, but a template of protocol parameters and I/O characterization that allows you to customize to emulate an application workload. In a Protocol Workload Model, key workload parameters can be varied such as % of read vs % of writes, % random vs % sequential, distribution of file or block sizes, etc.
 - Constant-Load Workloads: these are Protocol Workloads that generate constant Load (i.e. IOPS and Throughput) throughout the duration of the Workload. Constant-Load Workloads are recommended for those who want to understand the impact of external changes such as disabling controllers, optimizing configurations or changing firmware. By having a constant workload, it is easy to measure the impact of these changes. They are also useful to find the maximum sustained IOPs that can be achieved for a given workload.
 - Temporal Workloads: these are Protocol Workloads that generate that varies over time based on data acquired from a production environment throughout the duration of the Workload. Temporal Workloads are recommended for those who understand the impact of workload that vary over time. It is often useful to scale these workloads up or combine them with other workload for capacity planning purposes.
 - Example Workloads: these are Protocol Workloads with pre-defined protocol parameters and I/O characterization, based on a specific instance of a specific application in a specific deployment. The source of the pre-defined protocol parameters and I/O characterization is provided for each Example Workload. Example Workloads are recommended for those who do not have data about the production workloads they want to model, and want some examples to start with.
- Application Workloads: WorkloadWisdom-E also provides several highly intuitive Application Workloads, implemented based on specific applications analyzed by Virtual Instruments. Application Workloads differ from Protocol Workloads in that Application Workloads provide parameters that are specific to the Application, such as Number of VMs and Linked Clone Size for VDI Application Workloads, instead of protocol parameters such as % of read vs % of writes in the case of Protocol Workloads. Application Workloads are recommended for those who are familiar with the specific application they want to model, and want to do so using parameters specific to that application.
- TDE Workloads: where specific Protocol Workload Models or Application Workloads are not provided and the WorkloadWisdom appliance supports the protocol, an existing test authored in the WorkloadWisdom Test Development (TDE) environment can be imported and run in WorkloadWisdom. TDE Workloads are recommended for those who are storage protocol experts, and have a strong need to create customized workloads that are not provided by WorkloadWisdom.

Composite Workloads

Most applications are made up of several types of dissimilar workloads. For example, a database will often have some LUNs that are used to store the data and others to archive changes to the data. Often the data LUNs will be more than 90% read traffic while the archive LUNs will have more than 90% write traffic. There are also often other differences such as the amount of data requested in each transaction. Rather than creating a workload that averages these differences into a single workload, it is best to maintain this diversity. Averaging or combining dissimilar workloads generally leads to a workload that doesn't match any of the workloads in the environment and will often lead to misleading results.

Composite workloads make it easy to combine run multiple dissimilar workloads as part of the same test, using the same ports, and obtain aggregated results for the overall impact of these workloads. Some application environments such as web servers will show more commonality and can be used more broadly than other applications like databases, which can differ greatly from implementation to implementation.

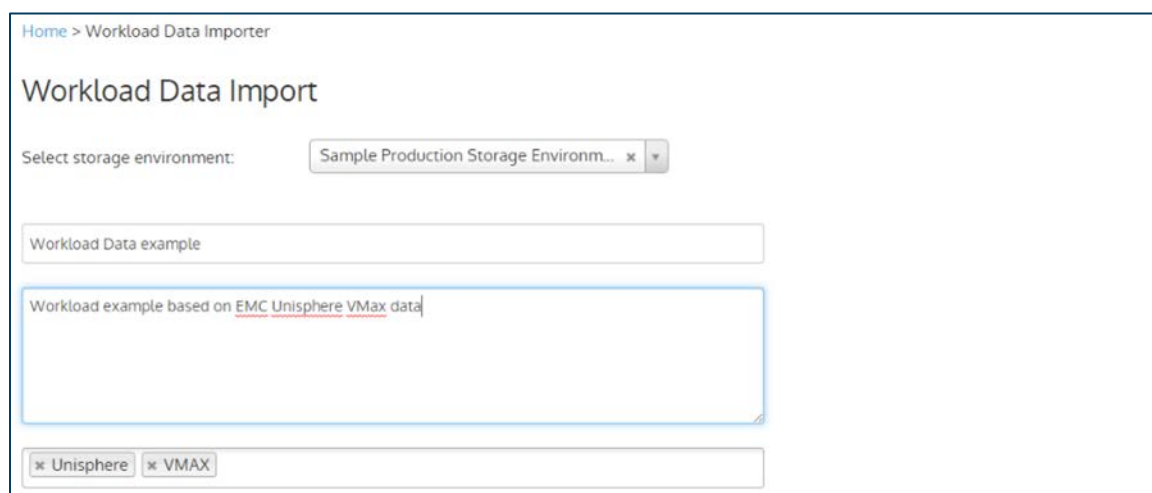
The Workload Acquisition Analyzer automatically combines like workloads while keeping workloads that are dissimilar separate. It provides a confidence level indicating how accurately it is able to combine workloads while still maintaining the diversity of dissimilar workloads. Increasing the number of workloads can improve the confidence level. Through the use of Analysis Policies users can choose which characteristics are important to maintain diversity and which ones are less critical when combining workloads.

Workload profiles from the storage subsystem via Workload Data Importer

Commercial storage arrays collect much of the information necessary to fully characterize both the Access Patterns as well as the Load Patterns (except temporality is usually measured in minutes rather than seconds). These statistics are often available through utilities that typically allow collection of over 1 min to 5 min intervals for each LUN/Volume. Output is usually in the form of a CSV file. You may need to consult with your storage vendors to determine the best approach to obtain both the Access Pattern information as well as the Load Patterns if you are not already familiar with how to acquire these statistics. We will add this level of assistance in future versions of this paper.

Workload Data Importer

The days of spending dozens of hours parsing log files, in the often futile attempt to create realistic workload models are over. The Workload Data Importer module of WorkloadWisdom is designed to import and analyze workload data from storage infrastructures, automatically, to create extremely realistic workload models. The workload data can be exported from storage vendor monitoring tools or server utilities such IOStat as Comma Separated Variable (CSV) text files. The data is analyzed to detect workload IO profiles to understand the workload characteristics in terms of IOPs, Throughput, Read/Write ratio etc.



The screenshot shows the 'Workload Data Import' web interface. At the top, there is a breadcrumb 'Home > Workload Data Importer'. The main heading is 'Workload Data Import'. Below this, there is a 'Select storage environment:' label followed by a dropdown menu showing 'Sample Production Storage Environm...'. Underneath is a text input field containing 'Workload Data example'. Below that is a larger text area containing 'Workload example based on [EMC Unisphere VMax data](#)'. At the bottom, there are two tags: 'Unisphere' and 'VMAX', each with a close button (x).

The first step is to choose a Production Storage Environment you want to associate the data with. This allows you to import multiple workload data sets and associate them with the same infrastructure. If you don't have a Production Storage Environment configured, you can use the pre-defined sample Production Storage Environment.

Importing a Workload Data File

The Workload Data Importer feature of WorkloadWisdom is designed to import, process, and analyze CSV based text files. It's capable of parsing, extracting values and calculating new values from a combination of values if required.

Each row in the data needs to represent a distinct time sample, with all metrics represented as columns within the CSV file. The columns do not need to be in a specific order.

To simplify the process WorkloadWisdom provides multiple Analysis Policies for different vendor's data. When you upload data it will automatically attempt to recognize the format and recommend a set of policies that match the data format. An Analysis Policy is essentially a "conversion" template that converts the imported data into the specific set of metrics that WorkloadWisdom requires to perform analysis. The metrics required to perform analysis include, but not limited to, Read / Write percentages, Read / Write IO sizes, and IOPS.

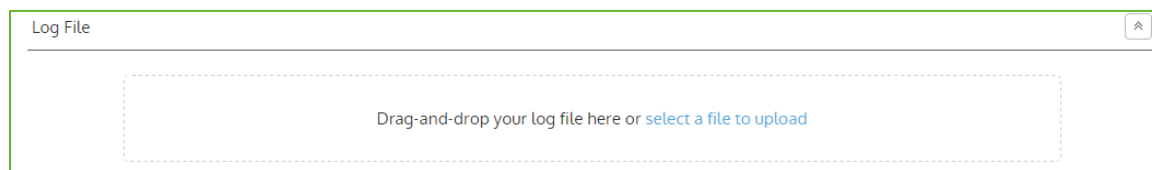


Analysis Policies are designed to be extensible and customizable. Therefore, you can either edit an Analysis Policy so that it can interpret the Input Data from the .csv, or you can edit the import .csv column labels to so that it can be interpreted by an available Analysis Policy.

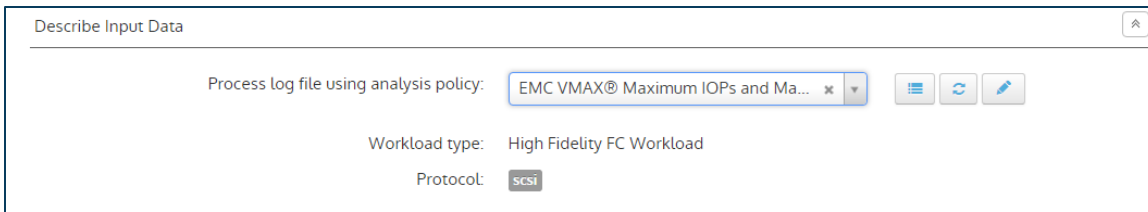
As different storage arrays keep track of workload data differently, and produce .csv in different structures, the Analysis Policy also provides mechanisms to perform calculations on the data in the imported .csv to derive the metrics WorkloadWisdom requires to perform analysis on the .csv. For example, if a storage array provides data on Throughput and IOPS, but does not provide data on average Block Size, then the Analysis Policy can derive the average Block Size by dividing Throughput by IOPS.

As an example, here is a truncated screenshot of an imported .csv from a storage array:

Date and Time	LDEV Number	Read I/O /sec	Write I/O /sec	Read Response Rate	Write Response Rate	Total Response Rate	Read Hit %	Read Xfer /sec
11 03 2014 00:10:00	0:00:00	0	0	0	0	0	0	0
11 03 2014 00:10:00	0:00:01	0	0	0	0	0	0	0
11 03 2014 00:10:00	0:00:02	0	0	0	0	0	0	0
11 03 2014 00:10:00	00:00:F0	0	0	0	0	0	0	0
11 03 2014 00:10:00	00:00:F1	0	0	0	0	0	0	0
11 03 2014 00:10:00	00:00:F2	0.42666668	0	47.1675	0	47.1675	100	0
11 03 2014 00:10:00	0:01:00	0	0	0	0	0	0	0
11 03 2014 00:10:00	0:01:01	0	0	0	0	0	0	0
11 03 2014 00:10:00	0:01:02	0	0	0	0	0	0	0
11 03 2014 00:10:00	0:02:00	2.12666668	255.42	607.0736	79.22789	83.586525	95.454544	0.5133333
11 03 2014 00:10:00	0:02:01	0.19333333	3260.7834	666.86896	76.45593	76.49094	100	0.036666665
11 03 2014 00:10:00	0:02:02	1.61	252.52333	605.2114	73.59288	76.96082	94.82401	0.3833333
11 03 2014 00:10:00	0:02:03	0	250.88	0	73.79562	73.79562	0	0
11 03 2014 00:10:00	0:02:04	0	0	0	0	0	0	0
11 03 2014 00:10:00	0:02:05	0	0	0	0	0	0	0
11 03 2014 00:10:00	0:02:06	0	0	0	0	0	0	0
11 03 2014 00:10:00	0:02:07	0	0	0	0	0	0	0
11 03 2014 00:10:00	0:03:00	113.48	255.36667	71.07204	78.851234	76.45787	99.95594	1.2
11 03 2014 00:10:00	0:03:01	70.67	3263.6934	944.1221	76.21483	94.60964	94.78798	17.173334
11 03 2014 00:10:00	0:03:02	111.36667	252.65	69.80611	73.68148	72.49586	99.95211	1.0533333
11 03 2014 00:10:00	0:03:03	110.74333	251.12666	66.59224	73.63113	71.47701	100	1.0066667



The Importer will upload the file and filter the list of Analysis Policies to those that match the file. There is an Analysis Policy for each storage array type or vendor utility; it defines the required and optional data required for the workload model.



Matching the file to the Analysis Policy, in this case to the EMC® VMAX®

WorkloadWisdom provides several policies out of the box for storage vendor monitoring tools. Contact your Virtual Instruments representative to discuss the addition of Analysis Policies.

Editing Analysis Parameters

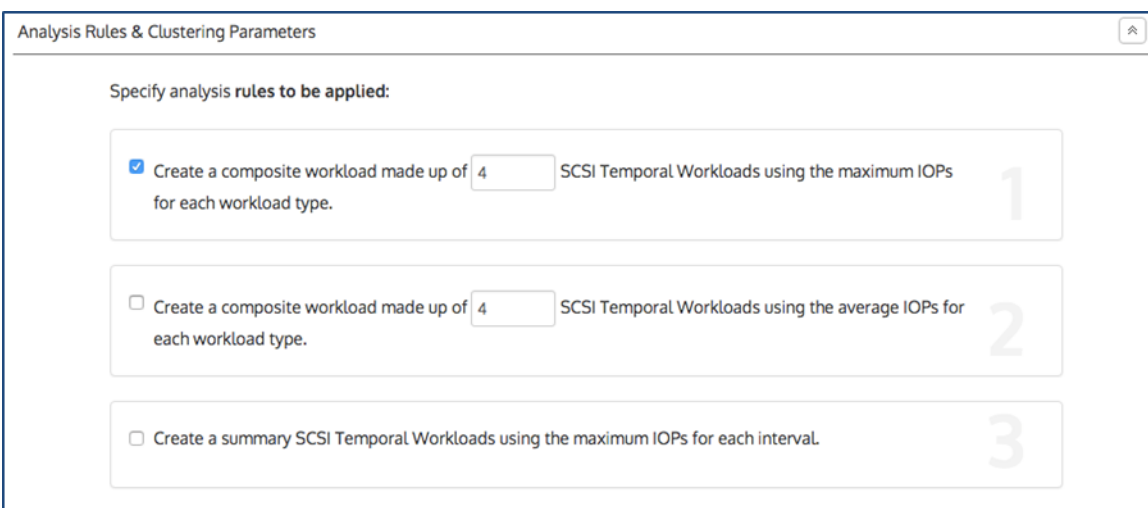
For parameters that are available from the vendors' solutions, Analysis Parameters allow you to specify values to be used in creating workloads. For example, average block sizes can be typically calculated by dividing throughput by Read/Write operations. Applications use a variety of block sizes so Analysis Parameters allow you to specify the block sizes typically configured in the environment. The analysis can automatically determine a distribution of these block sizes to arrive at the average observed by the vendors' solutions. The types of parameters that are user settable are defined in the Calculation Rules Analysis Policy which specifies what user input is required to support calculated values.

Parameter	Default value	Units
Server Block Size 1	4KB	N/A
Server Block Size 2	8KB	N/A
Server Block Size 3	64KB	N/A
Server Block Size 4	256KB	N/A

Setting new values

Specifying Workloads to Auto-Create

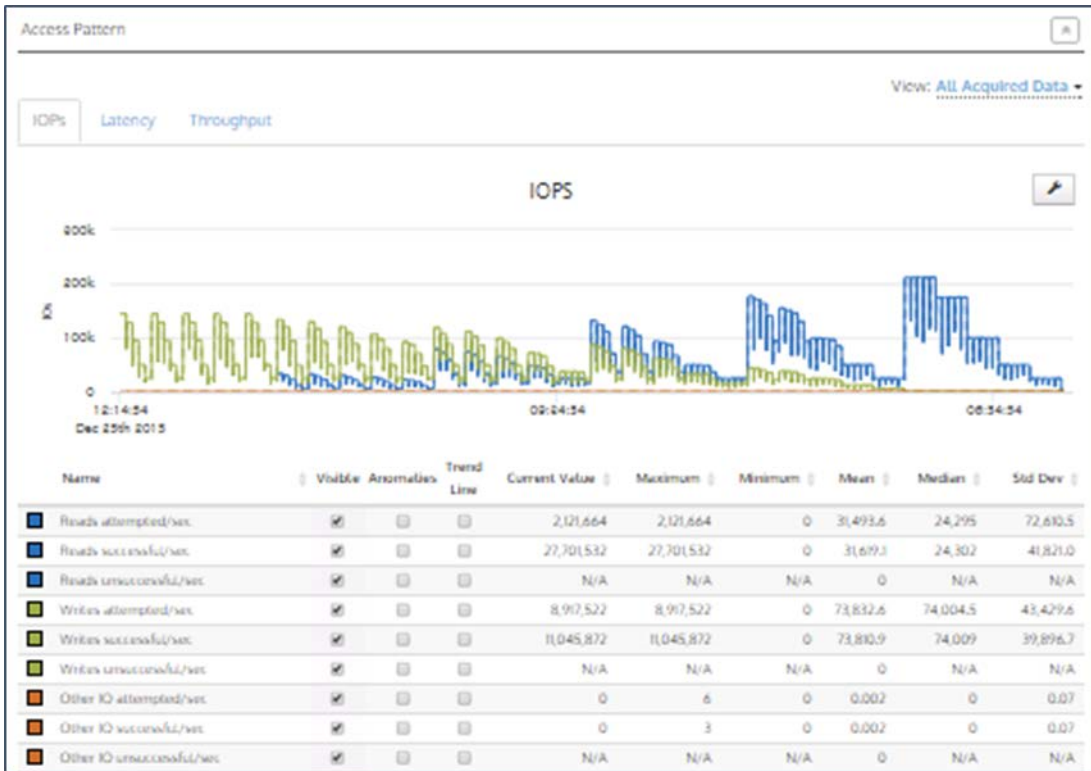
The Analysis Policy will normally specify which workloads to automatically create as well as ones that can optionally be created. You can choose to manually create individual workloads directly within the Workload Data Importer. Your workloads are now ready to run on the WorkloadWisdom Workload Generation Appliance.



Specifying analysis rules

Visualizing the Production Workload Data using Workload Data Importer

On the Production Workload Data screen, you can visualize the workload's IO Profile like IOPs, throughput, latency, read/write ratio, command mix, block size etc.



Production Workload Data, visualizing various I/O profile metrics

During a Workload Data Import, the data will be displayed in real-time on the charts as it is parsed.

Workload Data Importer Analysis Results

The Analysis Results section of the Production Workload Data is similar to that of the ones acquired from an Acquisition Profile via a VirtualWisdom Probe. The exception is that some of the charts may be empty if the array or the Analysis Policy lacked the information.

Summary View

Workload Components View

Throughput (MB)

1.6k

Read: 793
Write: 763

IOPs

48k

Read: 53k
Write: 48k

Latency (ms)

0

Read: 0
Write: 0

Errors

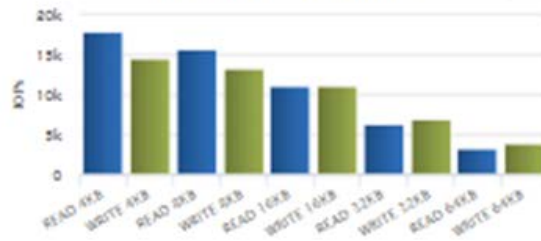
0

Read: 0
Write: 0

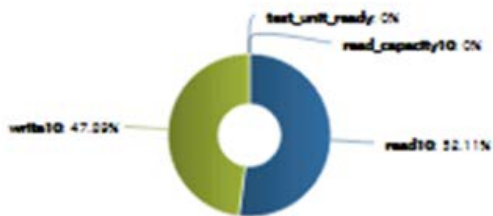
Read / Write Mix



Request Size Distribution



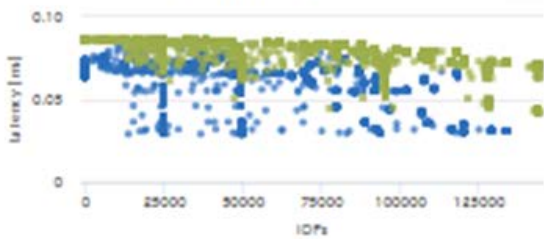
Command Mix



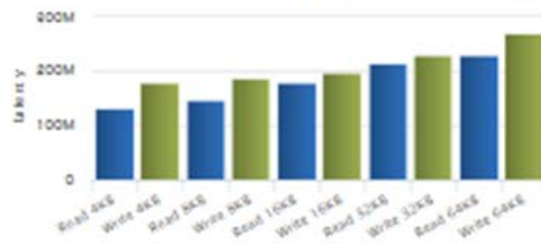
Average Request Size



Latency vs IOPs



Latency vs Request Size



Workload Analysis

Workload Data Importer using data from the VW hardware probes

Workload Data Importer can create a very effective workload model from the rich set of data available from the VirtualWisdom hardware probes. The data obtained from FC probes includes:

1. Date / time
2. LUN
3. Host I/Os per sec
4. Target
5. Initiator
6. % reads
7. % writes
8. Avg read size (can be merged with avg write size as min req'd)
9. Avg write size (can be merged with avg read size as min req'd)
10. Block size / request size
11. Latency for writes (can be merged with 'successful')
12. Latency for reads (can be merged with 'successful')
13. Latency for other
14. Latency vs. IOPS (calculated)
15. Avg clustering quality (calculated)
16. Reads attempted / sec
17. Reads successful / sec
18. Reads unsuccessful / sec
19. Writes attempted / sec
20. Writes successful / sec
21. Writes unsuccessful / sec
22. Other I/O attempted / sec
23. Other I/O successful / sec
24. Other I/O unsuccessful / sec
25. Latency for successful reads
26. Latency for successful writes
27. Latency for successful other
28. Read throughput
29. Write throughput
30. Other throughput

VirtualWisdom 4.X (and later releases) directions for exporting the CSV file:

Connect to an external VW instance and login as Administrator/admin to extract all of the Initiator-Target-LUN data for the host to create an output in .CSV format.

“extract-vw-probefc-to-Idx.py” extracts data from the VI Hardware Probe (ProbeFC) to CSV

Syntax is: “python extract-vw-probefc-to-Idx.py -v 10.10.10.10 -u Administrator -p admin -e HOSTNAME -o output.csv -t 2016-04-06T12:00:00/2016-04-06T18:00:00”

This will connect to an external VW instance with IP address 10.10.10.10 and login as Administrator/admin (alternatively the password can be put in a file which can be passed with -z if the user doesn't want to pass a

password on the command prompt, help can be found by running the script with no parameters) .. It will extract all of the Initiator-Target-LUN data for the host named 'HOSTNAME' from noon to 6pm on 4/6/16. Output format looks like:

```
Time,Command,Device,IO Size (KB),IOPS,Throughput (MB/s),Latency (ms)
2016-03-09 13:00:00,Read,SVCS_UCS15_A:VNX9759_A5:0,16.0,2.0,0.031,0.478
2016-03-09 13:00:00,Write,SVCS_UCS15_A:VNX9759_A5:0,6.1,15.0,0.093,0.885
2016-03-09 13:00:00,Other,SVCS_UCS15_A:VNX9759_A5:0,0.0,0.0,0.000,0.000
2016-03-09 13:01:00,Read,SVCS_UCS15_A:VNX9759_A5:0,16.0,2.0,0.038,0.514
```

Note: VI is also using similar scripts with a few customers that need to integrate VI data into their own custom tools as well

Workload Profiles using data from the VW ProbeVM (for vSphere)

Workload Data Importer can create a very effective workload model from the rich set of data available from the VirtualWisdom ProbeVM software probes. We can analyze and cluster workloads by Datastore which should build a good workload. In the case where there is 1 to 1 LUN to Datastores it is a better source in many cases than the array data. The data obtained includes:

1. Date / time
2. LUN
3. Host I/Os per sec
4. Target
5. Initiator
6. % reads
7. % writes
8. Avg read size (can be merged with avg write size as min req'd)
9. Avg write size (can be merged with avg read size as min req'd)
10. Latency for writes (can be merged with 'successful')
11. Latency for reads (can be merged with 'successful')
12. Reads successful / sec
13. Writes successful / sec
14. Read throughput
15. Write throughput

Storage Infrastructure Vendor Examples

The following section contains information on what metrics can and should be made available from various vendor storage arrays. Note that there might be changes based on array firmware releases and the various tool releases. It's important to note that we can create profiles for any vendor; the profiles below are simply the first ones we've completed. Contact your Virtual Instruments representative and we will create one for you.

EMC VNX UNISPHERE®

EMC's VNX Unisphere outputs CSV files that can be used to determine both Access Patterns and Load Patterns within the array. These can be run by your storage administrator or by the system engineer from EMC. Note that if there are more metric types in the file than are listed, they will just be ignored. Metrics in grey cells are required; others are optional and may be used. Example snippet below, based on a block workload, which may not exactly mirror your CSV file:

DATE	IN-STANCE	Host IOs/sec	% Reads	% Writes	Avg Read Size (KB)	Avg Write Size (KB)	Total Capacity (GB)	% Random Writes	% Seq Writes	% Random Reads	% Seq Read
11/25/2015 12:05	01E7C	34.52	93.22	6.78	8.01	16.75	201	6.33	0.45	92.42	0.8
11/25/2015 12:05	01G19	31.37	95.31	4.69	8.18	28.54	201	4.58	0.11	94.67	0.65
11/25/2015 12:05	033DC	0.64	24.24	0	0.48	0	21	0	0	22.47	1.77
11/25/2015 12:05	033B0	0.43	17.58	0	0.44	0	21	0	0	16.62	0.96
11/25/2015 12:05	033B4	2.94	65.53	34.47	40.26	46.14	101	32.65	1.81	63.04	2.49
11/25/2015 12:05	033B8	6.33	88.57	11.43	23.36	74.88	101	10.22	1.21	85.41	3.16
11/25/2015 12:05	033BC	14.25	90.1	3.34	16.71	22.75	101	2.91	0.44	88.52	1.57
11/25/2015 12:05	033C0	1.67	43.38	19.13	27.49	22.29	101	18.5	0.63	41.63	1.75
11/25/2015 12:05	033B4	9.9	81.99	18.01	25.62	52.56	101	15.66	2.36	81.01	0.98
11/25/2015 12:05	033C8	19.41	96.5	3.5	15	9.04	101	3.49	0.02	96.36	0.14

How to extract CSV block workload files from VNX option 1 ¹

Tool: EMC Unisphere

Note: these instructions will give you a different file than the one we show above.

1. In top Unisphere toolbar, select the storage subsystem from the drop down field on the far left
2. Select "system" in that same toolbar
3. Under "archive management", you can retrieve an archive, listed as 'nar' files
 - a. The default of 1-minute granularity gives you 2.5 hours of recorded data for each nar file
 - b. Ensure that the time granularity is identical for the entire time period
4. In "save file as" select where you want the file to be
5. Find the date/time you want, select 'retrieve', and when it's complete, click 'done'.
6. If you want to download multiple nar files, under 'archive mgmt', select 'merge archive' and follow the steps to select multiple files and keep merging until you have the time period you need/want.
7. Go to 'dump archive', open your merged nar file, click 'next' until ...
8. You see Object options. Here, select "all LUNs" or just leave it alone, and click 'next'

9. Select 'comma' as your field delimiter, then 'new line', then quoting character = 'none', then 'local time' and click 'next then 'finish'
10. If you get a message about preferences not persisting, ignore it.
11. Navigate to your new csv file and view it to confirm it exists. Do not open the file with Excel though you will be tempted to. Excel changes the time format and this doesn't match the format our Analysis Policy.

How to extract CSV workload files from VNX option 2 – step 3 is different ¹

Tool: EMC Unisphere

Note: these instructions will give you a different file than the one we show above.

1. In top Unisphere toolbar, select the storage subsystem from the drop down field on the far left
2. Select "system" in that same toolbar
3. Under "Monitoring and Alerts" then "Statistics for Block", you can merge or dump an archives, listed as 'nar' files
 - a. The default of 1-minute granularity gives you 2.5 hours of recorded data for each nar file
 - b. Ensure that the time granularity is identical for the entire time period
4. In "save file as" select where you want the file to be
5. Find the date/time you want, select 'retrieve', and when it's complete, click 'done'.
6. If you want to download multiple nar files, under 'archive mgmt', select 'merge archive' and follow the steps to select multiple files and keep merging until you have the time period you need/want.
7. Go to 'dump archive', open your merged nar file, click 'next' until ...
8. You see Object options. Here, select "all LUNs" or just leave it alone, and click 'next'
9. Select 'comma' as your field delimiter, then 'new line', then quoting character = 'none', then 'local time' and click 'next then 'finish'
10. If you get a message about preferences not persisting, ignore it.
11. Navigate to your new csv file and view it to confirm it exists. Do not open the file with Excel though you will be tempted to. Excel changes the time format and this doesn't match the format our Analysis Policy.

EMC VMAX UNISPHERE®

Depending on the type of storage array, EMC's VMAX Unisphere outputs CSV files that can be used to determine both Access Patterns and Load Patterns within the array. These can be run by your storage administrator or by the system engineer from EMC. Note that if there are more metric types in the file than are listed, they will just be ignored. Metrics in grey cells are required; others are optional and may be used. Example snippet below for block workloads:

DATE	INTER VAL (min)	INSTA NCE	Host IOs /sec	% Reads	% Writes	Avg Read Size (KB)	Avg Write Size (KB)	Total Capacity (GB)	% Ran dom Writes	% Seq Writes	% Ran dom Reads	% Seq Read
12/13/2015 16:10	5	0071G	26.8	0.01	99.99	0	8.39	100.12	97.88	2.11	0.01	0
12/13/2015 16:15	5	0071G	23.25	0.04	95.84	0.01	8.19	100.12	93.33	2.5	0.04	0
12/13/2015 16:20	5	0071G	28.16	0.15	99.85	0.05	8.73	100.12	97.56	2.28	0.15	0
12/13/2015 16:25	5	0071G	24.75	0.4	99.6	0.71	8.55	100.12	97.31	2.29	0.4	0
12/13/2015 12:30	5	0071G	5.12	16.74	83.26	18.8	8.11	522.05	77.46	5.8	16.29	0.46
12/13/2015 12:35	5	0071G	2.13	4.81	36.75	0.79	2.42	522.05	34.48	2.27	4.61	0.19
12/13/2015 12:40	5	0071G	2.56	1.63	37.4	0.38	2.39	522.05	33.74	3.66	1.52	0.1
12/13/2015 12:45	5	0071G	5.31	24.66	75.34	17.7	9.58	522.05	71.4	3.94	24.28	0.38
12/13/2015 12:50	5	0071G	2.35	4.69	39.25	1.14	2.67	522.05	36	3.25	4.38	0.31
12/13/2015 12:55	5	0071G	1.49	1.09	26.09	0.43	1.34	522.05	23.97	2.12	0.91	0.18
12/13/2015 13:00	5	0071G	3.83	14.72	85.28	12.3	7.52	522.05	78.48	6.79	14.46	0.26
12/13/2015 13:05	5	0071G	1.49	3.28	23.91	0.41	1.29	522.05	21.18	2.73	3.22	0.06
12/13/2015 13:10	5	0071G	2.35	1.93	41.96	0.48	2.13	522.05	38.59	3.37	1.87	0.06
12/13/2015 13:15	5	0071G	4.48	15.4	84.6	20.4	9.09	522.05	79.02	5.58	14.88	0.52

How to extract CSV block workload files from VMAX ¹

Tool: EMC Unisphere for VMAX (v8.0.x and later)

1. In top Unisphere toolbar, select the Performance section
2. Edit the report
3. Select the Symmetrix ID, specify CSV, and pick the time range, start and end, and timezone
4. Select "edit" query 1
5. Under Category and Instances, select "array" as your category, then you will pick the storage group volumes
6. Select "next" and on the next screen (metrics), pick the metrics (select all) and select "finish" then "OK" to finish editing the report. Don't worry about picking the right metrics, if Workload Data Importer doesn't need a metric, it will ignore it. So select all.
7. Ensure that the time granularity is identical for the entire time period

8. "Run now" and "OK" to save, will create the CSV
9. Navigate to your new csv file and view it to confirm it exists. Do not open the file with Excel though you will be tempted to. Excel changes the time format and this doesn't match the format our Analysis Policy.

EMC Isilon

EMC's Isilon CLI outputs CSV files that can be used to determine both Access Patterns and Load Patterns within the array. These can be run by your storage administrator or by the system engineer from EMC. Note that if there are more metric types in the file than are listed, they will just be ignored.

Commands to extract the CSVs from Isilon systems ¹

```
isi nfs exports -f csv list
```

```
isi smb shares -f csv list
```

Ensure that the time granularity is identical for the entire time period

Navigate to your new csv file and view it to confirm it exists. Do not open the file with Excel though you will be tempted to. Excel changes the time format and this doesn't match the format our Analysis Policy.

HDS

HDS Tuning Manager provides much of the information necessary to determine Access Patterns and Load Patterns for HDS arrays. It displays average block sizes and latency measurements based, in some cases, by application types. Below are the metrics gleaned via Tuning Manager. Note that if there are more metric types in the file than are listed, they will just be ignored. Metrics in grey cells are required; others are optional and may be used. Example snippet below:

Date and Time	LDE V Number	Read I/O /sec	Write I/O /sec	Read Response Rate	Write Response Rate	Total Response Rate	Read Hit %	Read Xfer /sec	Write Xfer /sec	Sequential Total I/O /sec	Sequential Total Xfer /sec	Random Total I/O /sec	Random Total Xfer /sec
104201600:10:00	0:24:00	195.26334	52.17667	340.00034	94.24151	299.9492	9.808588	1.2933333	0.2933333	2.27	0.02	195.17	1.566667
104201600:10:00	0:24:01	196.26666	48.13333	339.39294	92.0433	303.5968	9.693264	1.3	0.2466667	0.9	0.00666667	193.5	1.536667
104201600:10:00	0:24:02	194.86333	50.96667	339.4463	93.91167	300.6198	10.77256	1.2866666	0.2733333	1.74	0.01666668	194.09	1.543333
104201600:10:00	0:24:03	195.87334	49.46	336.2149	95.1647	299.8599	11.65146	1.2933333	0.2733333	2.1066666	0.02	193.22667	1.546667
104201600:10:00	0:24:04	195.35333	57.28667	337.47516	94.12151	303.0051	10.93618	1.29	0.24	0.7833334	0.00666667	191.85667	1.523333
104201600:10:00	0:24:05	198.74	49.86333	334.5485	92.57019	298.163	11.83675	1.32	0.2733333	2.7533333	0.02333333	195.85	1.57
104201600:10:00	0:24:06	195.81	48.32333	333.61197	94.04577	298.6601	12.257	1.3233334	0.25	1.04	0.01	193.09334	1.56
104201600:10:00	0:24:07	197.13333	49.93	335.00027	94.17612	298.4239	11.90866	1.3066666	0.27	1.4533333	0.01333334	195.61	1.563333

There are two methods of exporting this data, from HDS VSPs running block workloads.

Method one [1](#):

Tool: [Export tool](#), an internal, non-commercial tool

Also used by NETAPP Insight

Steps from SVP.

1. Bring up Export tool on Windows or UNIX
2. Enter target IP address

3. Enter User ID
4. Enter password
5. Specify range. Ensure that the time granularity is identical for the entire time period.
 - a. Short = 44 hours at 1 minute polling intervals
 - b. Long = 72 hours at 15 minute polling intervals
6. Specify output path
7. Select LDEV files
8. Concatenate individual output files into a single CSV file. If you edit your files, use a text editor, not Excel.
9. Navigate to your new csv file and view it to confirm it exists. Do not open the file with Excel though you will be tempted to. Excel changes the time format and this doesn't match the format our Analysis Policy.

Method two [1](#):

Tool: HDS Tuning Manager

1. Bring up Tuning Manager
2. Select target array
3. Select Performance Reporter
4. Select and execute LDEV Performance Summary Report for all Control Units (CUs)
5. Export the CSV files
6. Concatenate files together. Concatenate individual output files into a single CSV file. If you edit your files, use a text editor, not Excel.
7. Navigate to your new csv file and view it to confirm it exists. Do not open the file with Excel though you will be tempted to. Excel changes the time format and this doesn't match the format our Analysis Policy

HPE 3PAR

HPE's 3PAR system outputs CSV files that can be used to determine both Access Patterns and Load Patterns within the array. These can be run by your storage administrator or by the system engineer from HPE. Note that if there are more metric types in the file than are listed, they will just be ignored.

Commands to extract the CSVs from HPE 3PAR systems for block workloads ¹

Tool: system cli

1. Run command using the following syntax, with real data shown for illustration purposes:
 - a. `srstatv lun -attime -etsecs "2016-04-05 12:00:00" -groupby VV_NAME`
 - b. This would be for the 5 minute interval for the volumes beginning at 12:00:00. Ensure that the time granularity is identical for the entire time period.
 - c. To add additional 5 minute intervals, you would need to run the command an increment the time, as in "d" below:
 - d. `srstatv lun -attime "2016-04-05 14:00:00" -etsecs "2016-04-05 12:05:00" -groupby VV_NAME`
2. Gives you time, R/W IOPS, R/W throughput, R/W size, LUN
3. See your output and view it to confirm everything worked.
4. Save or redirect each output it to an ascii/CSV file and concatenate.
5. Make educated guess on sequential/random stats to build your model

Tool: SSMC (Main 3PAR GUI mgmt tool)

1. Select these choices
 - a. Reports
 - b. Create report
 - c. Select (for system name)
 - d. Exported volumes – performance statistics
 - e. Name the report
 - f. Select objects (pick the virtual volumes you want to report on)
 - g. Add objects
 - h. Set time interval you want. Ensure that the time granularity is identical for the entire time period.
 - i. Select Date Range
 - j. Under Chart Options, check 'group by' vv name and uncheck host name if it's checked
 - k. Select 'hi res' under "sampling"
 - l. Select all the metrics in "show chart"
 - m. Create
 - n. Actions ... export to CSV
2. Navigate to your new csv file and view it to confirm it exists. Do not open the file with Excel though you will be tempted to. Excel changes the time format and this doesn't match the format our Analysis Policy.
3. Make educated guess on sequential/random stats to build your model

HPE EVA

HPE StorageWorks outputs CSV files that can be used to determine both Access Patterns and Load Patterns within the array. These can be run by your storage administrator or by the system engineer from HPE. Note that if there are more metric types in the file than are listed, they will just be ignored.

Commands to extract the CSVs from HPE EVA systems running block workloads ¹

Tool: HPE StorageWorks Command View EVA

1. Execute the following command to gather data and redirect it to a file: `evaperf all -csv -cont <no of sec> -dur <no of sec> -fo <filename>asdf`
2. Ensure that the time granularity is identical for the entire time period
3. Navigate to your new csv file and view it to confirm it exists. Do not open the file with Excel though you will be tempted to. Excel changes the time format and this doesn't match the format our Analysis Policy.
4. Activity is reported separately for each controller accessing a virtual disk. The total activity for each virtual disk is the sum of the reported activity for each controller. A virtual disk may also be a snapshot, snapclone, or a DR group member. In the output, logical unit number (LUN) is used interchangeably with virtual disk. The counters are:

Read Hit Req/s—The number of read requests per second completed from the array cache memory.

Read Hit MB/s—The rate at which data is read from the array cache memory because of read hit requests.

Read Hit Latency—The average time it takes to complete a read request (from initiation to information receipt) from the array cache memory.

Read Miss Req/s—The number of read requests (per second) that failed to complete from the array cache memory and were completed from physical disks instead.

Read Miss Data Rate—The rate at which data is read from physical disks because the data was not present in the array cache memory.

Read Miss Latency—The average time it takes to complete a read request (from initiation to information receipt) from the physical disks.

Write Req/s—The number of write requests per second completed to a virtual disk that were received from all hosts.

Write Data Rate—The rate at which data is written to the virtual disk by all hosts and includes transfers from the source array to the destination array.

Write Latency—This average time it takes to complete a write request (from initiation to receipt of write completion).

Flush Data Rate—The rate at which data is written to a physical disk for the associated virtual disk.

Mirror Data Rate—The rate at which data travels across the mirror port to complete read and write requests to a virtual disk.

Prefetch Data Rate—The rate at which data is read from the physical disk to cache in anticipation of subsequent reads when a sequential stream is detected.

IBM® XIV

XIV systems provide a very rich set of metrics by LUN. Block sizes are tracked by ranges (>512KB, 64-512KB, 8-64KB, and 0-8KB) which is good enough granularity. Note that if there are more metric types in the file than are listed, they will just be ignored. Metrics in grey cells are required; others are optional and may be used. Example of CSV column headers below:

Metrics (CSV column headers)
Time
Volume
Read Hit Very large - IOPS
Read Hit Very large - Latency
Read Hit Very large - Throughput
Read Hit Large - IOPS
Read Hit Large - Latency
Read Hit Large - Throughput
Read Hit Medium - IOPS
Read Hit Medium - Latency
Read Hit Medium - Throughput
Read Hit Small - IOPS
Read Hit Small - Latency
Read Hit Small - Throughput
Read Miss Very large - IOPS
Read Miss Very large - Latency
Read Miss Very large - Throughput
Read Miss Large - IOPS
Read Miss Large - Latency
Read Miss Large - Throughput
Read Miss Medium - IOPS
Read Miss Medium - Latency
Read Miss Medium - Throughput
Read Miss Small - IOPS
Read Miss Small - Latency
Read Miss Small - Throughput
Write Hit Very large - IOPS
Write Hit Very large - Latency
Write Hit Very large - Throughput
Write Hit Large - IOPS
Write Hit Large - Latency
Write Hit Large - Throughput
Write Hit Medium - IOPS

Write Hit Medium - Latency
Write Hit Medium - Throughput
Write Hit Small - IOPS
Write Hit Small - Latency
Write Hit Small - Throughput
Write Miss Very large - IOPS
Write Miss Very large - Latency
Write Miss Very large - Throughput
Write Miss Large - IOPS
Write Miss Large - Latency
Write Miss Large - Throughput
Write Miss Medium - IOPS
Write Miss Medium - Latency
Write Miss Medium - Throughput
Write Miss Small - IOPS
Write Miss Small - Latency
Write Miss Small - Throughput
Read Memory-Hit Very large - IOPS
Read Memory-Hit Very large - Latency
Read Memory-Hit Very large - Throughput
Read Memory-Hit Large - IOPS
Read Memory-Hit Large - Latency
Read Memory-Hit Large - Throughput
Read Memory-Hit Medium - IOPS
Read Memory-Hit Medium - Latency
Read Memory-Hit Medium - Throughput
Read Memory-Hit Small - IOPS
Read Memory-Hit Small - Latency
Read Memory-Hit Small - Throughput

IBM XIV - using the CSV file ¹

For building a WorkloadWisdom model using Workload Data Importer, use version 7.2 of the XIV Performance Reporter.

1. Download the XIV Performance Report version 7.2.zip from:
<https://www.ibm.com/developerworks/community/files/app?lang=en#/file/d280337f-8525-49a8-b48c-c081b35b4e72>
2. Open the zip file; you will see 10 files
3. Open xiv_setup.txt in an ascii editor
 - a. The line "save_csv=True" is commented out. Remove the comment to ensure the results are in csv
 - b. Comment the next line "save_csv=False"
 - c. In the next section that starts "Edit this list of IP/names ..." follow the directions and replace "report1" with the name of your target array, add the array IP address, and add your admin name and password

- d. In the section that starts “To disable collecting...” ensure that “get_volumes = 1 and the other lines are all = 0
4. Save the file to the same directory
5. Open the README.txt in an ascii editor and follow directions
 - a. Make sure you have a connection to the XIV box
 - b. You’ve just edited the xiv_setup.txt file in step 3, so you’re good
 - c. Run xiv_create_perf_report.bat
 - d. Open the resulting csv file in an ascii editor (NOT Excel) and see that you’ve captured what you need. Ensure that the time granularity is identical for the entire time period.
 - e. If a file is created for each volume, go to step 6
6. The volume files need to be appended with an additional column for the volume.
 - a. In the directory where all the files for each volume are, run the command:
 - i. On Windows: `for /f %a in ('dir /b *.csv') do for /f "tokens=*" %b in (%a) do echo %a,%b >> all.csv`
 - ii. On Mac: `for csv in *.csv; do awk -v csv="$csv" '{print csv,""$0}' $csv>>all.csv; done`
 - b. Open the all.csv files and change the first header in the first line to “Volume”. If you edit your files, use a text editor, not Excel.
 - c. Do a global search on “ and replace with an empty string

Optionally: globally search and replace any part of the file name that is not part of the volume name with an empty string for example “_output.csv” may be appended and isn’t needed. Navigate to your new csv file and view it to confirm it exists. Do not open the file with Excel though you will be tempted to. Excel changes the time format and this doesn’t match the format our Analysis Policy.

NetApp® FAS

There are several ways to obtain the information that NetApp gathers about how it is being accessed. Their NetApp Data ONTAP command line interface offers several commands that can be used to access real time data. For example, sysstat can provide information about each protocol down to a one second granularity. By collecting and aggregating the data over time both Access Patterns and Load Patterns can be determined. Or you can use Perfstats. Load your perfstat.zip, that represents the data captured during the business cycle you want to emulate, into the LatX Web to parse the log files. From there, you can export the stats into a CSV file, which you can then manipulate. For CDOT systems, make sure you've enabled QoS before capturing the log file. Note that if there are more metric types in the file than are listed, they will just be ignored. Metrics in grey cells are required; others are optional and may be used. Example of CSV column headers below:

Metrics (CSV column headers)
Iteration
Volume
Aggregate
Total Op/s
Read /s
Read Data (B/s)
Avg Read Size
Avg Read Response
Avg Read Concurrency
Avg Read Inter-Arrival Wait Time (MS)
Write /s
Write Data (B/s)
Avg Write Concurrency
Avg Write Inter-Arrival Wait Time (MS)
Other /s
Avg Other Response Time (ms)
Sync Free/s
DF Worker /s
ZMSG in use
Zombie Blks

Tool: [Export tool](#), an internal, non-commercial tool ¹

1. Bring up Export tool on Windows or UNIX
2. Enter target IP address
3. Enter User ID
4. Enter password
5. Specify range. Ensure that the time granularity is identical for the entire time period
 - a. Short = 44 hours at 1 minute polling intervals
 - b. Long = 72 hours at 15 minute polling intervals
6. Specify output path
7. Select LDEV files
8. Concatenate individual output files into a single CSV file. If you edit your files, use a text editor, not Excel.
9. For file protocol, select the files for the protocol you are interested in (CIFs, NFS, etc.) and concatenate them into a single file
10. Using excel or similar tool find the average percent distribution for each of the meta commands. This information can be entered into the analysis screen when the Volumes data is imported and analyzed

Pure Storage®

The Pure Storage script requires the Pure Storage python library (REST client) which can be obtained from GitHub <https://github.com/purestorage> or pip.

Pure Storage arrays collect data automatically and have 30 second interval data for up to 3 hours. They can collect data for up to a year but at longer intervals. This script pulls the historical data, and can either pull just from the current information or run and collect it over time. This script and parameters should work with either Python 2.7 or 3.X.

Usage: `Pure_Get_All_Volume_IO_Stats.py [options] ...` ¹

The script uses the Pure Storage API to pull historical (option d) volume IO statistics or collect (option c) them over time. Please go to [this site](#), for the script and more useful information; starting with the 'readme' file.

Options:

Array Options:

- s ARRAY_ADDRESS Pure Storage routable name or IP address
- a API_TOKEN An API token instead of user name and password.
- p PASSWORD Pure password [default: pureuser]
- u USER Pure user name [default: pureuser]

Pull Historical Data:

- d DURATION Pull a duration of historical data already collected by the array. Use 1h, 3h, 24h, 7d, 30d, 90d, or 1y [default: 1h]

Collect Data:

- c COLLECT Collect 30 second interval data for up to 3 days (3d). Use 1h to 3d.
- i Include 3 hours of history to get to collection duration quicker.

Output Files:

- o OUTPUT Name of output file [default: pfa-output.csv]
- e ERROR_OUT Error output file which will cause the script to log errors rather than prompt for input. For use in unattended scripts.

Other Options:

- h, --help show this help message and exit
- v, --version show program's version number and exit

¹ *Workload data extraction procedures we provide are maintained on a best-effort basis. The procedures may change as vendors make modifications to their products. If you encounter issues, please contact us at 'marketing@virtualinstruments.com'*



Sales
Sales@virtualinstruments.com
1.888-522.2557

Website
virtualinstruments.com