Storage Capacity Management using NetApp Operations Manager

Version 1.0
Status: Ready

Authored by:
Adaikkappan Arumugam, Raja Shekar, Shridhar Bandekar, Saravanan Manickam

18 Sep 2011

Abstract

NetApp Operations Manager provides storage administrators with the centralized management, monitoring, and reporting tools to manage and optimize NetApp storage. NetApp Operations Manager helps storage administrators resolve problems faster and improve capacity utilization by providing a full picture of NetApp storage resources.

To efficiently manage storage capacity, storage administrators require tools to view current utilization of the resources, change in utilization over a period of time, trend and forecast for utilization in future, charge users for the capacity utilized and alert administrators to identify and resolve imminent problems. This technical report describes the various tools provided by NetApp Operations Manager for capacity management.

Operations Manager Architecture

To understand the tools provided by Operations Manager for capacity management, it is important to know the architecture of the software and various modules. The knowledge of architecture will help in understanding the data collection mechanism and semantics of the data presented by Operations Manager.

Operations Manager includes management server software and a web interface. Server software installed on a Windows or Linux server consists of a database server, apache web server and a set of Operations Manager services - *monitor, server, eventd, scheduler and watchdog*. Operations Manager web interface provides the tools required for monitoring and reporting on NetApp storage. The tools can also be accessed using command line interface on the Operations Manager server.

monitor is a multi-threaded service responsible for periodically monitoring the storage system and its components. It is divided into ***monitor modules***, each responsible for collecting and saving specific attributes of a storage component to Operations Manager database. For example, cpuMon collects cpu usage of storage systems.

A monitor module runs on a storage object and collects attributes it is responsible for. For example, vFilerMon runs on a storage system collecting information about all vFiler units on the storage system, dfMon runs on a storage system collecting disk free space on volumes and aggregates on the storage system. Each monitor module has a ***monitoring interval*** that determines the frequency at which the monitor module runs and updates the attributes it is responsible for in Operations Manager database. monitor service schedules monitor modules to run on different objects at different times to distribute the activity uniformly over a period of time. The next run of the monitor on the same storage object is scheduled when the duration defined by the monitoring interval expires.

Operations Manager Reports

Operations Manager presents the current storage system configuration and capacity utilization in reports. Reports contain the information in the Operations Manager database at the point of report generation. Operations Manager provides a set of default reports and allows the storage administrator to define custom reports choosing attributes of various storage objects. Operations Manager does not provide reports that present information at an earlier point of time. Administrators can schedule the reports to be saved at periodic intervals if required.

The scheduling mechanism used by monitor service (described earlier) can have the following visible impact on reports: in a report that includes attributes of multiple storage objects, different attributes could be collected at different time. For example, in a report that includes volume and aggregate capacity, volume capacity could be from a different time and aggregate capacity could be from a different time.

Operations Manager History Graphs

Operations Manager saves the capacity utilization and performance collected in every monitoring cycle in history tables of database. The history is presented in Operations Manager UI as graphs such as "Volume Capacity Used". The graphs are also available at the server console using "dfm graph" CLI. Graphs are available for the object whose usage or performance is being tracked, as consolidated graph at a higher level storage container or as consolidated graph at a resource group.

Each graph can be viewed for different time periods - 1 day, 1 week, 1 month, 3 month and 1 year. In Operations Manager UI, these time periods are provided as hyperlinks under the graph. In the CLI, a suffix to the graph name (-1d, -1w, -1m, -3m and -1y) chooses the time period. Based on the time period, graphs present data of a period starting at the current time and going back 24 hours, 1 week, 1 month, 3 month or 1 year based on the selected graph period.

The CLI also provides additional functionality that is not available in the Operations Manager UI:

```
dfm graph [ <options> ] <graph-name> <object>

   The options are
    -s <start-date>
    -e <end-date>
    -D <date-format>
    -F <output-format>
    -h <height>
    -w <width>
```

- Specify a custom time period using start-date and end-date
- Format the date output using date-format
- Chose an format for the output from among text, html, csv, xls, png and gif, and size of the image in case of png and gif output
- Graph data older than 1 year (yearly data is kept in Operations Manager database forever)

Note that the default output of dfm graph command does not print the column names. csv and xls output formats print the column names also.

For further details, refer to Operations Manager man pages.

Accumulation of Samples in Operations Manager Database

Operations Manager server keeps history for each graph in the database in five tables in differing granularity: daily, weekly, monthly, quarterly and yearly. These are then used to generate graphs for the different time periods.

For each database table, the Operations Manager server saves sample values for periods of the following duration:

- Each daily history sample covers 15 minutes.
- Each weekly history sample covers two hours.
- Each monthly history sample covers eight hours.
- Each quarterly history sample covers one day.
- Each yearly history sample covers four days.

The scheduling mechanism used by monitor service (described earlier) can have the following visible impact on history graphs: capacity utilization collected at different time for different storage objects does not allow for easy consolidation or aggregation. For example, volume capacity utilization at a storage system level would be difficult to consolidate as utilization of different volumes is collected at different time. In addition, regular sample collection may be disrupted either due to Operations Manager down time or storage system unavailability. This will cause the samples to be missing for some storage objects, further complicating data consolidation. To solve this, irrespective of the actual data collection time - samples in history are saved into pre-defined sample periods that are rounded to a 15 minute boundary. Sample periods always begin at midnight, so the daily sample periods are 12:00 to 12:15am, 12:15 to 12:30am, and so forth.

Different monitor modules run at different monitoring intervals, and do not follow the 15 minute sample period used to store data in graph tables. For a list of monitor modules and their monitoring intervals, refer to section ***Monitors and Graphs*** at the end. For daily history, monitor modules may collect samples at 15 minute intervals, shorter than 15 minute intervals or longer than 15 minute intervals. When a monitor module runs at intervals shorter than 15 minutes, multiple samples collected are consolidated into a single sample in the database. When consolidating multiple samples into one sample in daily history tables, or for consolidating the values for less granular history tables such as weekly or monthly tables, multiple sample values are averaged. Next section discusses the consolidation mechanism in more detail.

Weekly, monthly, quarterly and yearly history tables are populated from the data collected and saved by monitor modules in the daily history tables. As mentioned earlier, each sample in weekly history table covers two hours. So, all the 15 minute daily history samples that are collected during a two hour period are consolidated into one weekly sample. Similarly, monthly history table is populated from the weekly history samples and so on.

Monitor Modules of Different Monitoring Intervals – An Example

Before describing the accumulation of samples by monitor modules that run at different monitoring intervals, the database tables used to store the samples are described as an example.

cpuHistoryDay table stores the cpu utilization of storage systems and contains the following columns:

Column	Description
Cupid	database ID of a storage system
Timestamp	starting time of the sample period
sampleCount	Number of samples that are consolidated in to this sample period (more details below)
cpuBusyPercentIntervalSum	sum of cpu usage samples consolidated in to this sample period

qtreeHistoryDay table stores the volume and qtree capacity utilization and contains the following columns (and a few more):

Column	Description
qtreeId	database ID of a volume or aggregate
Timestamp	starting time of the sample period
qtreeKBytesUsed	used space of the qtree
qtreeKBytesLimit	total space of the qtree quota

Based on the monitoring interval, a monitor module may run once, several times, or never during a particular graph sample period of 15 minutes used in daily history tables.

CASE 1: Monitoring interval is 15 mins

Sample value collected by the monitor in every monitoring cycle is saved as is. An example of such a monitor is ifMon which collects network interface traffic.

CASE 2: Monitoring interval less than 15 mins

A running total of the sample values collected by the monitor in every monitoring cycle are saved along with a count of samples that have been added to the total.

For example, let us use the case of cpuMon which runs at a monitoring interval of 5 minutes by default. In the first monitoring cycle, cpuBusyPercentIntervalSum = sample1 and sampleCount = 1, and a new entry (row) is created in the cpuHistoryDay table. In the second monitoring cycle, cpuBusyPercentIntervalSum = sample1 + sample2 and sampleCount = 2.

Assuming the third sample is collected at a time not more than 15 minutes from first sample, cpuBusyPercentIntervalSum = sample1 + sample2 + sample 3 and sampleCount = 3.

The fourth monitoring cycle is more than 15 minutes after the first sample time. At this point, a new entry (row) is created for the sample; the earlier row is left as it was at the end of third monitoring cycle. When generating graphs, cpuBusyPercentIntervalSum and sampleCount are used to get average value for cpu usage. An example cpu usage graph output below:

Example 1: cpu-1d graph for a storage system

```
Timestamp                   CPU Used
Tue Apr 14 23:00:00 2009    5.7
Tue Apr 14 23:15:00 2009    9.1
Tue Apr 14 23:30:00 2009    5.4
Tue Apr 14 23:45:00 2009    4.9
Wed Apr 15 00:00:00 2009    4.3
Wed Apr 15 00:15:00 2009    11.5
Wed Apr 15 00:30:00 2009    12.4
Wed Apr 15 00:45:00 2009    12.7
Wed Apr 15 01:00:00 2009    11.4
Wed Apr 15 01:15:00 2009    6.3
...
```

CASE 3: Monitoring interval more than 15 mins

Sample value collected by the monitor in every monitoring cycle is saved with starting of the 15 minute sample period as the timestamp. Monitoring does not happen again in the same 15 minute period, and it may not happen for many more 15 minute periods that follow.

For example, let us use the case of qtreeMon which runs at a monitoring interval of 8 hours by default. In the first monitoring cycle, qtreeKBytesUsed = sample1, and a new entry (row) is created in the qtreeHistoryDay table.

For the next 8 or more hours no monitoring happens, so no new rows are recorded. In the second monitoring cycle, a new row is created qtreeKBytesUsed = sample2. Depending on the time at which the monitor ran, second sample could fall into any 15 minute period after the 8 hour period (as seen in second sample in the example below).

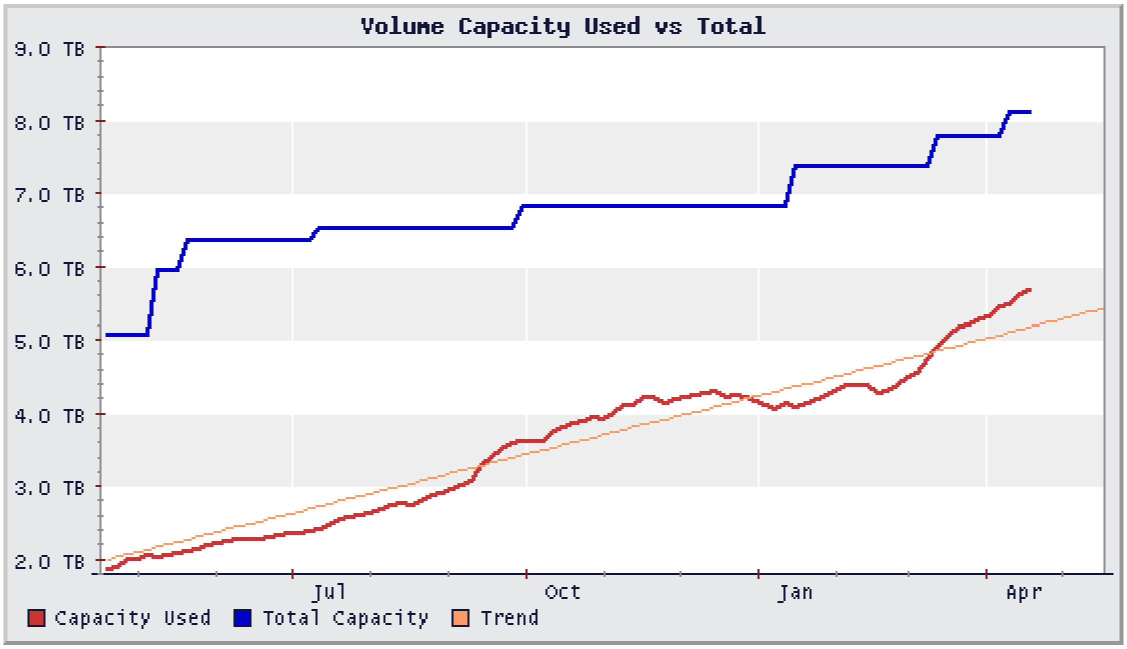
For monitors that have relatively long monitoring interval such as qtreeMon, the image for daily graph contains few data points. However they are connected to form a line which makes it appear as if there are more data points that are changing over time. Viewing the graph data in text, csv or xls format will show the actual data points collected.

Example 2: qtree-usage-1d graph for a qtree

```
Timestamp                  Capacity Used    Trend
Wed Apr 15 01:00:00 2009   11215511552      10409969168
Wed Apr 15 09:15:00 2009   11215511552      12851769244
Wed Apr 15 17:15:00 2009   16050290688      15219575378
Wed Apr 15 17:39:16 2009   -                15339281133
Wed Apr 15 18:03:32 2009   -                15458986888
Wed Apr 15 18:27:48 2009   -                15578692642
Wed Apr 15 18:52:04 2009   -                15698398397
Wed Apr 15 19:16:20 2009   -                15818104151
...
```

Trending in Operations Manager Graphs

Capacity utilization graphs in Operations Manager also include a trend line - a thinner and lighter line in the graph. The trend line is a linear regression of the graph line using the samples in the graph. Objective of performing a linear regression is to best-fit the data based on estimation. The trend line for graphs extends past the end of the actual data in the graph to offer an extrapolation of where the values might be if current trend continues.



Note: Trend is not available in all graphs.

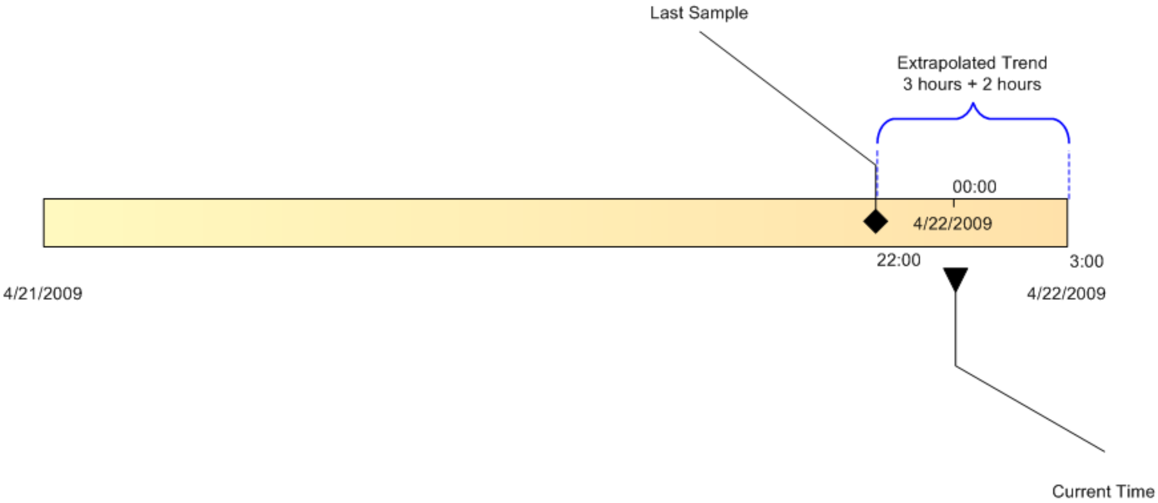
In textual graph outputs, trend data is included as the last column in the output. The output also extends past the end of the actual data for the graph, and such samples can be identified by "-" values for actual graph data (Capacity Used column in the qtree-usage-1d graph example above).

Note that the additional trend only samples need not be at the same sample interval as the actual data. Twenty additional trend samples are added to the graph, and they extend to a pre-defined time period based on the time scale of the graph.

- Daily graph contains trend for 3 hours in future.
- Weekly graph contains trend for a day in future.
- Monthly graph contains trend for 4 days in future.
- Quarterly graph contains trend for a week in future.
- Yearly graph contains trend for a month in future.

For example, in Qtree Capacity Used graph for one day, graph samples are found at 8 hour intervals. But the trend line continues for 3 more hours from the time at which graph is generated. Time interval between the last graph sample and (current time + 3 hours) is divided in to twenty parts and trend samples are added at these time points.

The diagram below describes the duration and distribution of time samples in a 1day graph. The last qtree space usage sample was collected at 22:00 hours on 4/21/2009. Current time is 00:00 hours on 4/22/2009. Since the daily graph contains trend 3 hours into the future, trend data spans 5 hours (from last sample to 3 hours in future). This 5 hour period is divided into 20 time slices and an extrapolated value is added for each time slice.



Purging of Older Samples from History Tables

To keep the database size under control, samples from each of the history tables are purged when they get old. A maximum of 150 samples are kept in each sample history table for one storage object, which translates into:

- 37.5 hours in daily sample table
- 12.5 days in weekly sample table
- 50 days in monthly sample table
- 5 months in quarterly sample table.
- Samples in yearly sample table are never purged.

Operations Manager UI does not provide graphs that span longer than a year; "dfm graph" cli can be used to get older data from the yearly sample table.

dfm graph CLI and Custom Time Periods

When using "dfm graph" cli to get graph samples for specific time periods, both the graph period suffix and the start-time/end-time inputs must be specified carefully.

To get last 24 hours of data (start-time = 86,400 and end-time = 0), if you use -1d suffix, you get the samples from the daily history table. This table provides the most granular data available (15 min samples at best based on the monitoring interval). However, if you specify -1w suffix, the data is provided from the weekly sample table in which granularity has already been reduced to 2 hour samples. Using longer time period graphs reduces the granularity of data even further.

Another consideration is the purge duration and the time period for which graph data is required. For example, to see the last 60 days of CPU usage for a particular storage system, use the "-3m" suffix, because "-1m" will not have all the data you request. It contains only last 50 days data.

Available Graphs

This table lists all the available graphs in Operations Manager, a brief description for each graph, columns in the graph and column description.

Graph Name	***Graph Description***	
	Columns	**Column Description**
cpu	*Graph of Storage System's CPU Utilization*	
	Timestamp	Sample time
	CPU Used	Operations Manager reports CPU usage as reported by ONTAP SNMP (cpuUpTime and cpuBusyTime). The ONTAP SNMP metric has improved over the releases to provide an accurate picture of cpu usage and headroom available. Releases prior to ONTAP 7.2 provided percentage of time any processor is busy as the CPU usage. Starting 7.2, the CPU usage also takes into account domains that don't have headroom or processors that don't have headroom. This includes processor utilization on multi-cpu systems.
filer-cpu	*An alias to "cpu" graph. Provides the same data as "cpu" graph.*	
Raw-capacity-used-vs-total	*Graph of used raw capacity and total raw capacity of disks attached to the storage system.*	
	Timestamp	Sample time
	Raw Capacity Used	Capacity of data and parity disks on the storage system.
	Raw Capacity Total	Capacity of all disks on the storage system.
Raw-capacity-used	*Graph of used raw capacity of disks attached to the storage system.*	
	Timestamp	Sample time
	Raw Capacity Used	Capacity of data and parity disks on the storage system.
Filer-nfs	*Graph of number of NFS operations served per second.*	
	Timestamp	Sample time
	NFS Operations/sec	NFS operations served per second.
filer-cifs	*Graph of number of CIFS operations served per second.*	
	Timestamp	Sample time
	CIFS Operations/sec	CIFS operations served per second.
filer-nfs-cifs	*Graph of number of NFS and CIFS operations served per second.*	
	Timestamp	Sample time

	NFS Operations/sec	NFS operations served per second.
	CIFS Operations/sec	CIFS operations served per second.
filer-http	*Graph of number of HTTP operations served per second.*	
	Timestamp	Sample time
	HTTP Operations/sec	HTTP operations served per second.
filer-fcp	*Graph of number of FCP operations served per second.*	
	Timestamp	Sample time
	FCP Operations/sec	FCP operations served per second.
filer-iscsi	*Graph of number of iSCSI operations served per second.*	
	Timestamp	Sample time
	iSCSI Operations/sec	iSCSI operations served per second.
vfiler-usage	*Graph of vFiler storage capacity usage.*	
	Timestamp	Sample time
	Capacity Used	Used capacity of all the volumes and qtrees directly owned by the vFiler unit. Qtree capacity is included only if a quota exists for the qtree.
	Trend	Used capacity trend.
qtree-usage-vs-total	*Graph of qtree space used, qtree quota and qtree space usage trend.*	
	Timestamp	Sample time
	Capacity Used	Used capacity of the qtree. Available only if quotas are enabled.
	Total Quota	Size of the qtree quota.
	Trend	Qtree capacity usage trend.
qtree-usage	*Graph of qtree space used and qtree space usage trend.*	
	Timestamp	Sample time
	Capacity Used	Used capacity of the qtree. Available only if quotas are enabled.
	Trend	Qtree capacity usage trend.
qtree-percent	*Graph of qtree space used as a percentage of qtree quota.*	
	Timestamp	Sample time
	Capacity Used	Qtree space used as a percentage of qtree quota.
user-disk-space-used	*Graph of disk space used by the user and usage trend.*	
	Timestamp	Sample time
	Disk Space Quota Used	Disk space used by the user.
	Trend	Disk space used trend.
user-disk-space-used-vs-total	*Graph of disk space used by the user, user quota hard limit and usage trend.*	
	Timestamp	Sample time
	Disk Space Quota Used	Disk space used by the user.
	Disk Space Quota Hard Limit	Disk space hard limit of the user quota.
	Trend	Disk space used trend.
user-disk-space-used-percent	*Graph of disk space used by the user as a percentage user quota hard limit.*	
	Timestamp	Sample time
	Disk Space Quota Used	Disk space used as a percentage of user quota hard limit.
user-files-used	*Graph of files used by the user and usage trend.*	

	Timestamp	Sample time
	Files Quota Used	Number of files owned by the user.
user-files-used-vs-total	*Graph of files used by the user, file quota hard limit and usage trend.*	
	Timestamp	Sample time
	Files Quota Used	Number of files owned by the user.
	Files Quota Hard Limit	Files hard limit of the user quota.
	Trend	Files used trend.
user-files-used-percent	*Graph of files used by the user as a percentage of the file quota hard limit.*	
	Timestamp	Sample time
	Files Quota Used	Files used by user as a percentage of files hard limit.
lun-bytes	*Graph of data read from and written to a LUN in bytes per second.*	
	Timestamp	Sample time
	Bytes Read/sec	Number of bytes read per second from lun
	Bytes Written/sec	Number of bytes written per second on lun
lun-ops	*Graph of total protocol operations per second performed on the LUN.*	
	Timestamp	Sample time
	Operations/sec	Number of operations per second performed on lun
snapmirror-lag	*Graph of lag time for SnapMirror destination volume or Qtree.*	
	Timestamp	Sample time
	SnapMirror Lag	Lag between the source and destination objects based on the last SnapMirror update time.
snapmirror-data-change	*Graph of amount of data transferred for every transfer that occurred from a SnapMirror source to a SnapMirror destination. As only the changed data is transferred in a SnapMirror transfer, the graph also represents the data change (new data since the last transfer) on the source volume.*	
	Timestamp	Sample time
	SnapMirror Data Change	Amount of data transferred in bytes for every transfer that occurred from a SnapMirror source to a SnapMirror destination.
snapmirror-bytes	*Graph of data transfer rate from SnapMirror source to SnapMirror destination.*	
	Timestamp	Sample time
	SnapMirror Transfer Rate	Data transferred in bytes per second.
network-bytes	*Graph of bytes read in and written out to the network per second.*	
	Timestamp	Sample time
	Network Bytes In/sec	Bytes read in per second
	Network Bytes Out/sec	Bytes written out per second
fc-bytes	*Graph of Fibre Channel traffic per second on the switch over time.*	

	Timestamp	Sample time
	RX Bytes/sec	Bytes read from the FC network per second
	TX Bytes/sec	Bytes written to the FC network per second
hba-port-bytes	*Graph of bytes read from and written to the HBA port per second.*	
	Timestamp	Sample time
	HBA Port Bytes In/sec	Bytes read from the HBA port per second.
	HBA Port Bytes Out/sec	Bytes written to the HBA port per second.
hba-port-frames	*Graph of frames received and transmitted by the HBA port per second.*	
	Timestamp	Sample time
	HBA Port Frames In/sec	Frames received by the HBA port per second.
	HBA Port Frames Out/sec	Frames transmitted by the HBA port per second.
volume-usage-vs-total	*Graph of amount of storage space used, total space of the volume and the usage trend.*	
	Timestamp	Sample time
	Capacity Used	Amount of storage space used by the volume, in kilobytes
	Total Capacity	Total capacity allocated to and used by the volume, in kilobytes
	Trend	Capacity usage trend.
volume-usage	*Graph of disk space used by the volume and the usage trend.*	
	Timestamp	Sample time
	Capacity Used	Amount of storage space used by the volume, in kilobytes
	Trend	Capacity usage trend.
volume-percent	*Graph of disk space used as a percentage of total space of the volume.*	
	Timestamp	Sample time
	Capacity Used	Amount of storage space used by the volume expressed in percentage to total
volume-overwrite-rate	*Graph of data overwrite rate per day for the volume*	
	Timestamp	Sample time
	Data Overwrite Rate	The volume data overwrite rate in kilobytes per day
	Trend	Overwrite rate trend
volume-space-reserve	*Graph of space currently available in the space reservation of the volume.*	
	Timestamp	Sample time
	Available Space Reservation	Overwrite reserve space available in the volume, in kilobytes
	Trend	Trend of available space reservation.
volume-total-space-reserve	*Graph of total size of space-reserved files on the volume*	
	Timestamp	Sample time

	Total Space Reservation	Total overwrite reserve space in the volume, in kilobytes
	Trend	Trend of total space reservation
volume-first-snapshot	*Graph of free space available on the volume after the first snapshot*	
	Timestamp	Sample time
	Space After First Snapshot	Available space in the volume after the first snapshot copy, in kilobytes
	Trend	Trend of space available after first snapshot copy.
aggregate-usage-vs-total	*Graph of aggregate space used, total space and usage trend.*	
	Timestamp	Sample time
	Capacity Used	Amount of used space in the aggregate, in kilobytes
	Total Capacity	Total capacity of the aggregate, in kilobytes
	Trend	Trend of capacity usage
aggregate-usage	*Graph of aggregate space used and usage trend.*	
	Timestamp	Sample time
	Capacity Used	Amount of used space in the aggregate, in kilobytes
	Trend	Trend of capacity usage.
aggregate-percent	*Graph of aggregate space usage as a percentage of aggregate total space.*	
	Timestamp	Sample time
	Capacity Used	Amount of storage space used in the aggregate as a percentage of total space
aggregate-usage-vs-committed	*Graph of aggregate space used and amount of committed aggregate space.*	
	Timestamp	Sample time
	Capacity Used	Amount of used space in the aggregate, in kilobytes
	Space Committed	Amount of the total aggregate space committed, in kilobyte
	Trend	Trend of used capacity.
aggregate-usage-vs-committed-percent	*Graph of aggregate used space percentage and committed aggregate space percentage.*	
	Timestamp	Sample time
	Percent Capacity Used	Amount of used space in the aggregate as a percentage of total space
	Percent Space Committed	Amount of the total aggregate space committed as percentage of total aggregate space

Monitors and Graphs

This table lists the monitor modules in Operations Manager that collect data for graphs, the monitoring interval for each module and the list of graphs for which the module collects data.

Monitor Module	***Monitoring Interval Option***	***Default Interval***
	Graphs Updated	**Graph Available For**
qtreeMon+	*qtreeMonInterval*	*8 hours*
	qtree-usage-vs-total qtree-usage qtree-percent	Qtree Resource Group
userQuotaMon+	*userQuotaMonInterval*	*1 day*
	user-disk-space-used user-disk-space-used-vs-total user-disk-space-used-percent user-files-used user-files-used-vs-total user-files-used-percent	Quota User
cpuMon	*cpuMonInterval*	*5 minutes*
	cpu filer-cpu	Storage System vFiler Unit Resource Group
filerOpsMon	*opsMonInterval*	*10 minutes*
	filer-nfs filer-cifs filer-nfs-cifs filer-http filer-fcp filer-iscsi	Storage System Resource Group
dfMon	*dfMonInterval*	*30 minutes*
	vfiler-usage	vFiler Unit
	volume-usage-vs-total volume-usage	Resource Group Storage System Volume
	volume-percent volume-overwrite-rate volume-space-reserve volume-total-space-reserve volume-first-snapshot	Volume
	aggregate-usage-vs-total aggregate-usage aggregate-usage-vs-committed	Resource Group Aggregate
	aggregate-percent aggregate-usage-vs-committed-percent	Aggregate
sanhostMon	*SANHostMonInterval*	*5 minutes*
	hba-port-bytes hba-port-frames	HBA Ports Resource Group
snapmirrorMon	*snapmirrorMonInterval*	*30 minutes*

	snapmirror-lag snapmirror-data-change snapmirror-bytes	SnapMirror Destination Volumes SnapMirror Destination Qtrees
lunMon	*lunMonInterval*	*30 minutes*
	lun-bytes lun-ops	LUN Resource Group
ifMon	*ifMonInterval*	*15 minutes*
	network-bytes	Storage System Resource Group
fcMon	*fcMonInterval*	*5 minutes*
	fc-bytes	FC Switch Resource Group

+ qtreeMon collects qtree capacity usage using SNMP when host login and password are not provided for the storage system. When host login and password are provided, userQuotaMon collects the qtree capacity usage using XML APIs.

Authors:

Raja Shekar, Shridhar Bandekar, Saravanan Manickam, Adaikkappan Arumugam
NetApp Inc

Change Log:

Version	Description
0.1	Initial draft covering Operations Manager history graphs

To Do:

1) Describe effect of manual discovery on graph samples
2) History Data in Operations Manager Database Access
3) Web UI names for graphs