Creating Baseline Performance Reports for EMC Symmetrix®

Charles Macdonald
Senior Technology Specialist
TELUS
charles.macdonald@telus.com

Table of Contents

Disclaimer: The views, processes or methodologies published in this article are those of the authors. They do not necessarily reflect EMC Corporation's views, processes or methodologies.

Introduction

Performance Management for storage arrays requires the collection and analysis of historical performance data, with the goal of anticipating growth, and detecting potential problem areas before they significantly impact the environment. We compare current performance to past performance to extrapolate growth. We must compare current data to historical data to determine the impact of changes in the storage environment.

Both of these functions require us to collect and manipulate historical data to provide an accurate baseline value for analysis. Baseline data is generally a collection of values collected over a reasonably large sample period, and then averaged. In addition to averages, we should include measures of availability in baseline statistics for performance analysis. EMC Performance Manager provides "maximum values" in addition to averages, but does not provide any other measure of variability. However, we can leverage the Performance Manager CLI (pmcli.exe) to produce customized, automated reports that use other measures of variability.

This article explains:

- what statistics we should include in a general purpose baseline collection for EMC Symmetrix arrays
- how to create standard deviation as a measure of variability for the baseline
- how to leverage the EMC Performance Manager CLI (pmcli.exe) to extract performance statistics for the baseline
- how to automate the creation of the baseline using commonly available tools

While this article focuses on Symmetrix, the techniques we discuss can be easily adapted for other Performance Manager objects, or for Navisphere® Analyzer.

Statistical Methods

The data sets used to generate baseline storage performance consist of many data points. To make sense of the data, we use a measure of *central tendency*, and a measure of data dispersion, called *spread*.

The measures of central tendency are the *mean*, *median*, and *mode*; they are intended to represent the expected value of a data set. To create performance baselines, we are only concerned with the *mean*, defined as:

$$\textit{mean} = \frac{\text{the } \textit{sum} \text{ of all points in the data set}}{\text{the } \textit{number} \text{ of data in the data set}}$$

Data points will be scattered on either side of the mean value. Measures of *spread* indicate how widely the sample data is scattered. We will use the *standard deviation:*

$$\textit{standard deviation} = \sqrt{\left(\frac{\text{the } \textit{sum} \text{ of the } \textit{square} \text{ of differences from the } \textit{mean} \text{ for all points in the data set}}{\text{the } \textit{number} \text{ of data points in the data set} - 1} \right)}$$

The standard deviation is expressed in the same units as the data, so can be represented on the same graph. For a data set with *normal distribution*, the standard deviation, 68% of the data points lie within one standard deviation from the mean. 95% of the data points lie within two standard deviations from the mean.

The ability of the sample mean to represent the mean of the *population* (the set of all data), is sensitive to the sample size, and to skewed data (*outliers)* in the sample set. The *standard error of the mean* approximates the variation in the mean value calculated from different sample sets of the population. It is calculated as follows:

$$\textit{standard error of the mean} = \frac{\text{standard deviation}}{\sqrt{\text{ of the } \textit{number} \text{ of data points in the data set}}}$$

The table below shows the standard error of the mean as a percentage of standard deviation for various sample sizes:

Table 1 – Standard Error of the Mean

Sample Size	Standard Error of the Mean as a % of Standard Deviation
1	100
7	37.8
14	26.7
21	21.8
30	18.3
60	12.9
90	10.5

30 to 90 days of data should provide a fairly accurate mean to create a storage baseline.

Note: Confidence Intervals and Confidence Levels are beyond the scope of this paper, but may be useful when comparing a baseline using recent data to a baseline using older data, e.g. to extrapolate growth.

The data in the Table 2 is System I/O per Second (IOPS) for a DMX3. We collected data points for 30 days, at 11:00 a.m. The mean and the standard deviation for this data set are calculated in the examples below.

Table 2 – Sample System I/O per Second at 1100 AM

Sample Day	IOPS	Sample Day	IOPS	Sample Day	IOPS
4075.103	7612.167	8453.146	9218.134	10311.965	11858.351
4191.797	7612.167	8618.47	9328.858	10985.139	11906.529
4346.53	7682.182	8701.437	9632.221	11038.66	12610.885
5095.87	7787.563	8779.135	9831.791	11757.391	12670.518
7014.45	8052.247	8853.777	10197.629	11764.78	14972.127

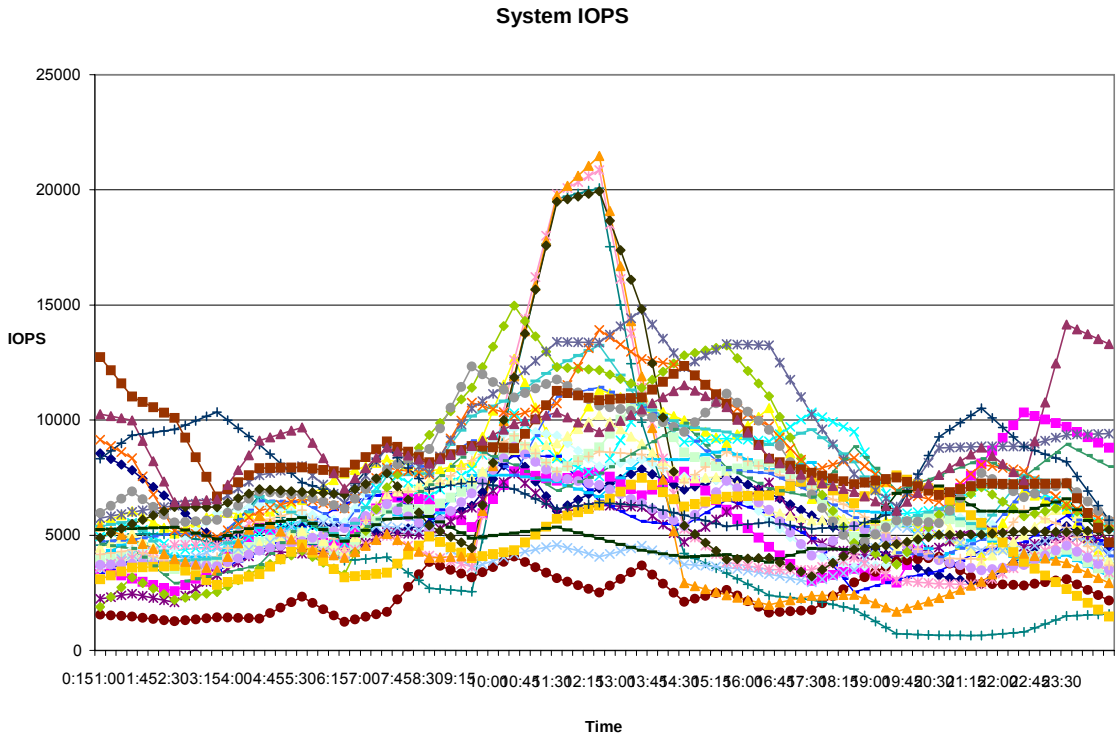
We can calculate the mean using the Microsoft Excel function AVERAGE(). The mean of the data in Table 2 is 9165 IOPS.

Calculate the standard deviation using the Microsoft Excel function STDEV(). The standard deviation of the data in Table 2 is 2642.

If you check the values in the table, you will find that 21 of the 30 values (70%) fall within one standard deviation of the mean.

EMC Performance Manager applied these methods to a collection of 30 days of data. The data points for each of the 30 days are in 15 minute intervals. The first graph shows all of the data points collected:

Figure 1 – System IOPS – 30 days



The second graph shows the calculated mean:

Figure 2 – System IOPS - Mean

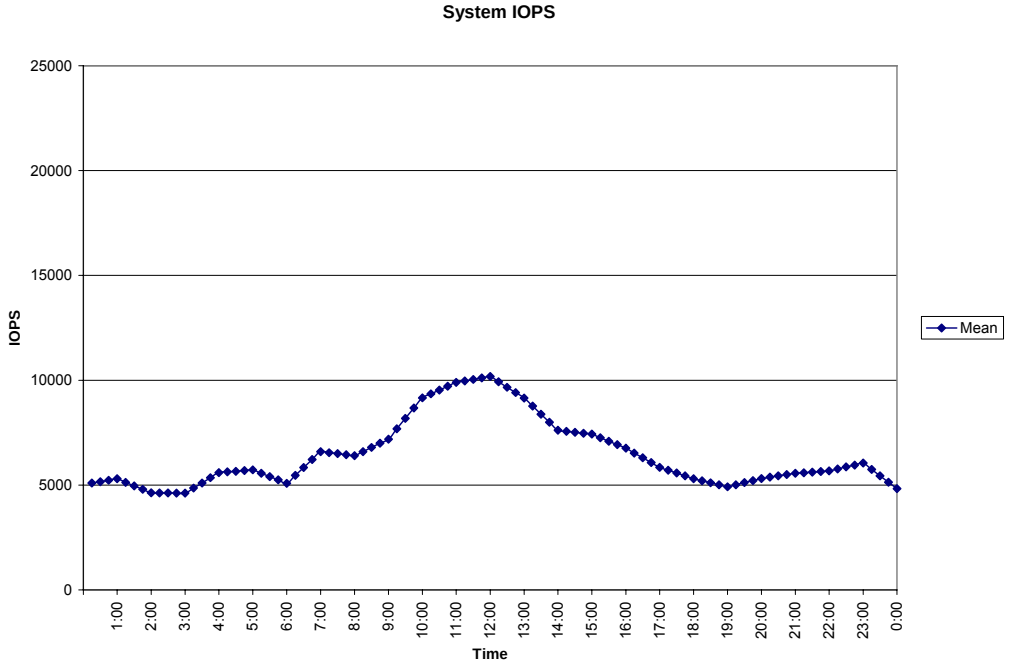


Figure 3 uses error bars in Excel to show one standard deviation from the mean.

Figure 3 – System IOPS – Mean and Standard Deviation

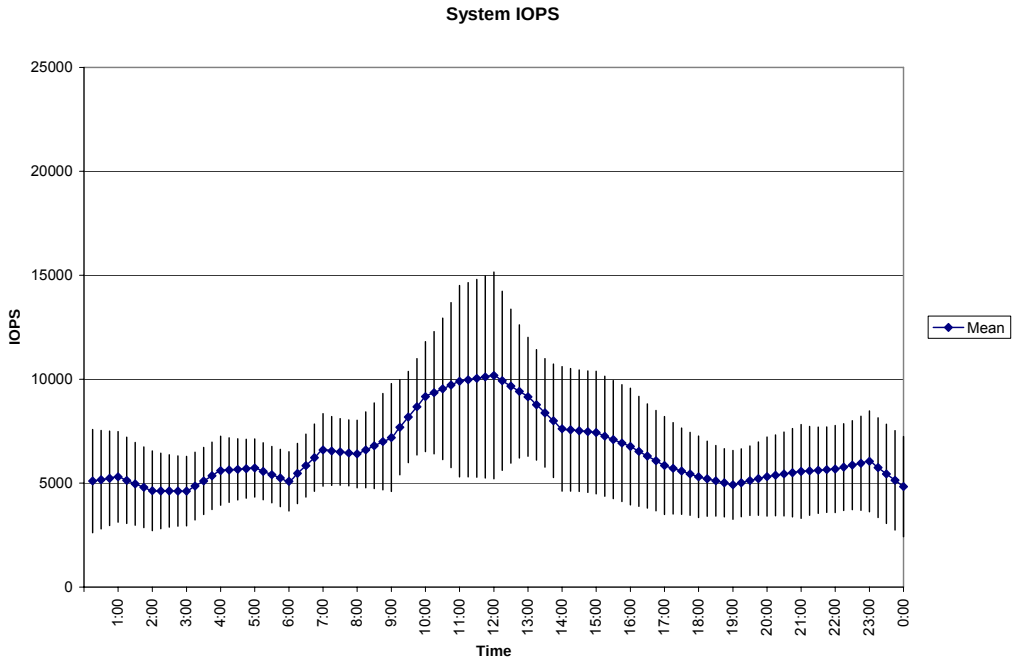
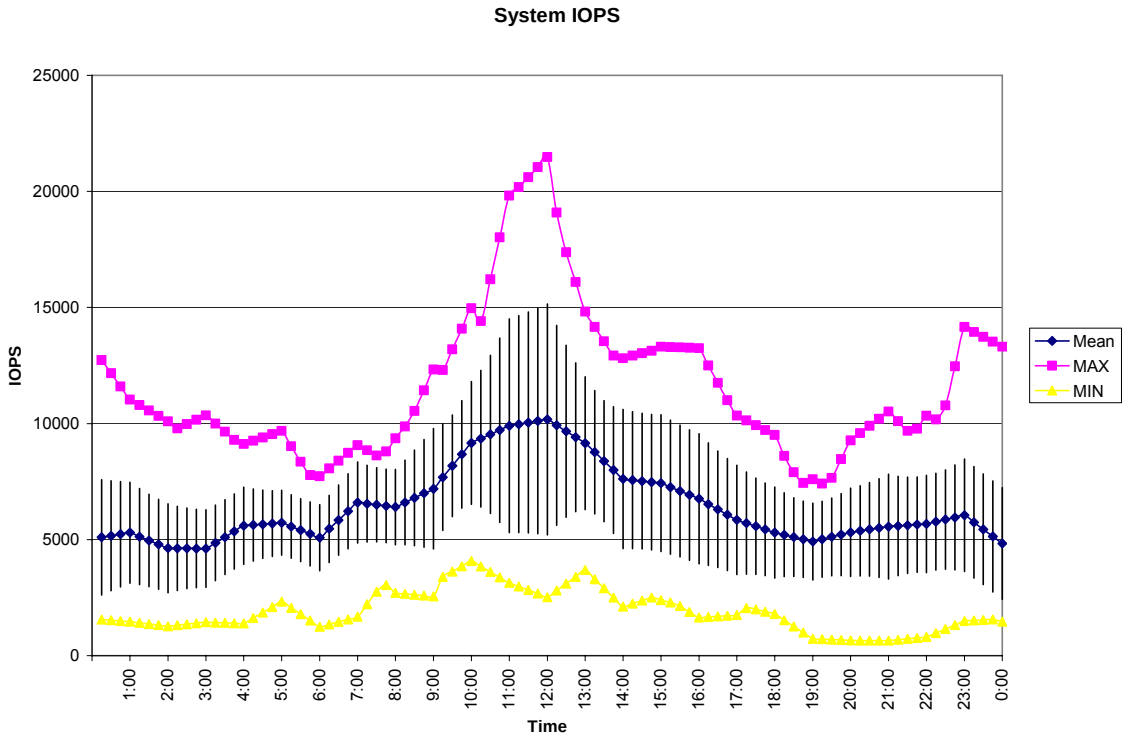


Figure 4 includes the minimum and maximum values for comparison.

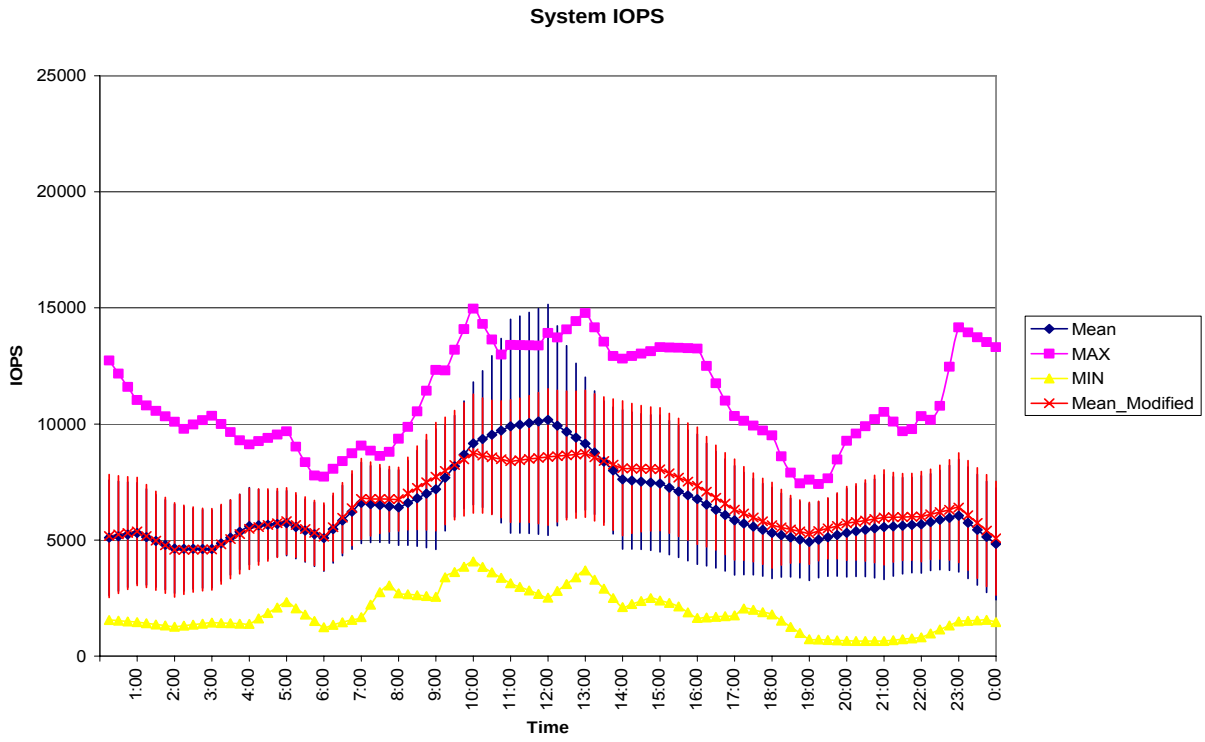
Figure 4 – System IOPS – Mean, Standard Deviation, Min and Max



The thirty days of data for the array included three days that had unusually high IOPS at noon (see figure 1). To illustrate how outlier data affects the mean, we removed the three days from the data set in Figure 5. Removing the three days has a large affect on the maximum IOPS shown in the graph, and flattens out the peak in the mean.

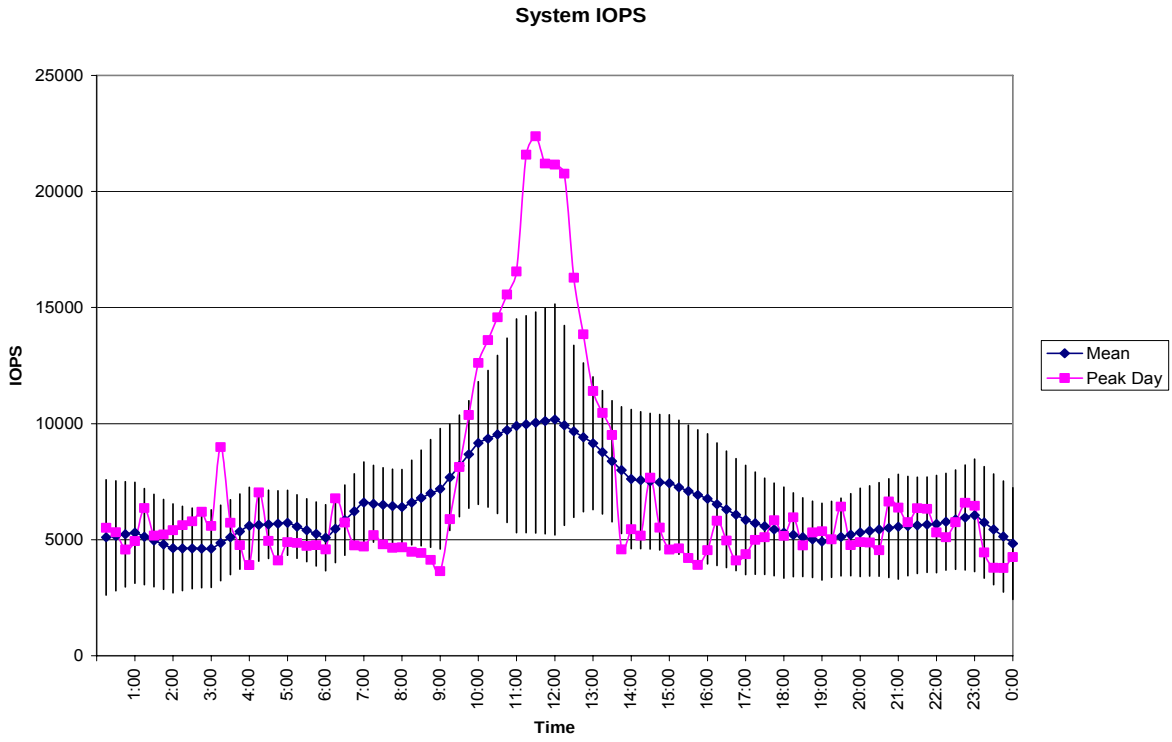
See Affect of Mean on Outlying Data on next page.

Figure 5 – Affect on Mean of Outlying Data



Finally, we chose one of the peak days for comparison to the baseline, as shown in Figure 6. The peak of the "Peak Day" line is outside of one standard deviation away from the baseline mean, and therefore represents a day that is in the top 16% for System IOPS during the period 1000 to 1400.

Figure 6 – Peak Day Compared to Baseline Mean



Creating a Baseline for Symmetrix Performance Data

EMC ControlCenter® Agents initially collect the performance data available to the EMC Performance Manager. In the case of Symmetrix, data is collected by the Storage Agent using the Symmetrix API.

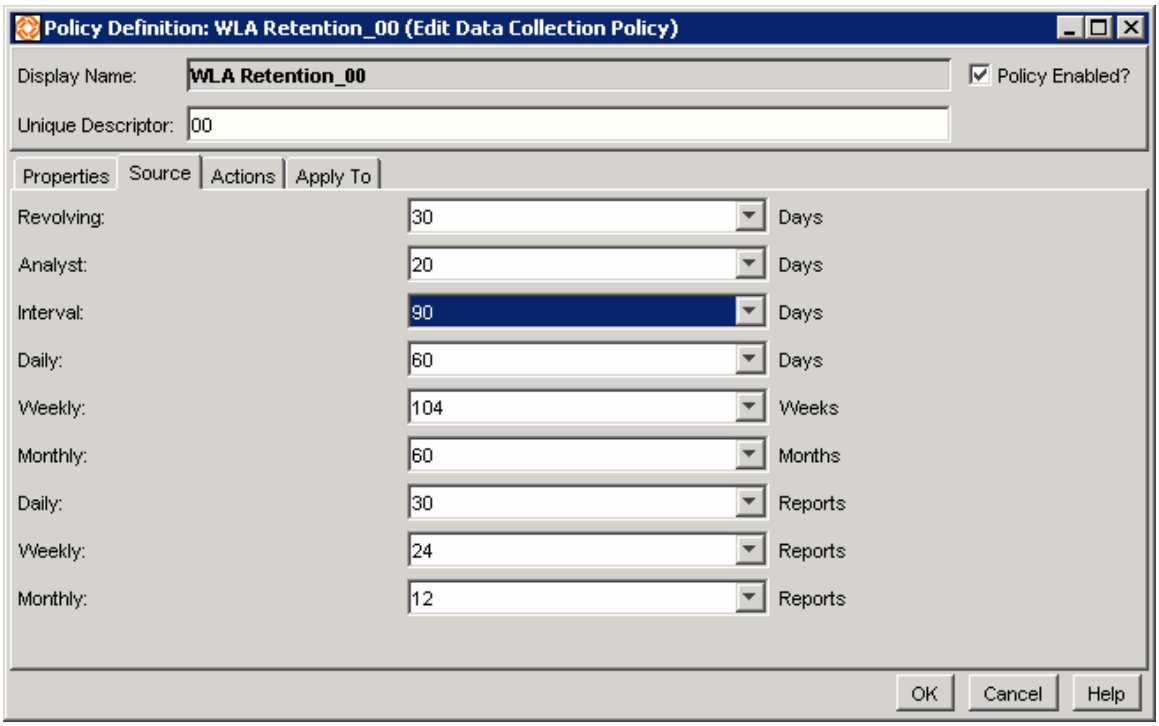
The Storage Agent for Symmetrix then passes the data to the Workload Analyzer (WLA) Archiver Agent. The WLA Archiver Agent initially stores data in temporary text files with a .ttp extension. When data collection is complete, the .ttp files are converted to a binary format, and stored as a .btp file. These initial files are *interval* files. The number of data points collected per hour in an interval file is determined by the configuration of the WLA Archiver Data Collection Policy for the Control Center agent. A 15 minute interval is adequate to create a baseline.

Interval .btp files are rolled into *daily* .btp files. Data in the interval files is averaged to create a data point for each hour in a daily file. For example, if the interval file has data in 15 minute increments, four incremental data points will be averaged to create an hourly data point for the daily file. W*eekly* .btp files are created by averaging the data points in seven daily files, and *monthly* .btp files are created from four weekly .btp files.

To create a baseline, the daily, weekly, and monthly .btp files are undesirable as they do not contain enough information to calculate the *spread* of the raw data.

Retention periods for .btp files are controlled by the WLA Retention policy in EMC ControlCenter®, found under Administration → Data Collection Policies → Policy Definitions → WLA Archiver. You must define a sufficient retention period for the Interval collections that allows you to use 30 or more days of data; Figure 7 shows a retention period of 90 days for Interval .btp files.

Figure 7 – WLA Retention Policy Settings



Key Metrics

EMC Performance Manager identifies Symmetrix metrics in six categories (or more if you are using SRDF, ESCON, etc.):

- Devices
- Dir-DA
- Dir-Fibre
- Dir-Port
- Disks
- System

For the baseline, it is convenient to regroup specific metrics into three broad categories:

- Front End
- Cache
- Back End

Metrics of particular interest for the baseline are described in the tables below. The metrics are also defined in the EMC Performance Manager Help files under “Metrics Glossary”:

Table 3 – Front End Metrics

Metric Category	Metric	Notes
Dir-Fibre	ios per sec	ios per sec for a director helps you to balance host io over the front end of the array.
Dir-Fibre	% util*	Directors are limited by bandwidth (iops). As % util approaches 100%, response time will suffer.
Dir-Port	% util*	Ports are limited by throughput (KBps). As % util approaches 100%, response time will suffer.

Table 4 – Cache Metrics

Metric Category	Metric	Notes
System	ios per sec	This metric is the sum of writes and random reads; it gives a good overview of the total host activity on the array. It does not include sequential reads; sequential reads are included in the metric "total ios per sec."
System	% hit	
System	% write	
System	Kbytes read per sec	This metric is for all Symmetrix devices
System	Kbytes written per sec	This metric is for all Symmetrix devices
System	read hits per sec	
System	system max wp limit	This metric is fixed at 80% of the available cache.
System	number write pending tracks	The number of tracks in cache that have not yet been destaged to disk. This metric should not approach the system max wp limit.
Dir-Fibre	% hit	
Dir-Fibre	slot collisions per sec	slot collisions cause misses
Dir-Fibre	% write	
Dir-Fibre	% read hit	Used to isolate changes in % hit
Dir-Fibre	requests per sec	Compare to Dir-Fibre ios per sec to check for possible cache misalignment
Dir-Port	average io size in Kbytes	If io size is larger than cache slot size, requests per io will increase.

Table 5 – Back End Metrics

Metric Category	Metric	Notes
Dir-DA	ios per sec	Helps plan for balancing the back end directors and planning growth
Dir-DA	% util*	Sustained high % util may indicate lack of balance or over-utilization
Dir-DA	prefetched tracks per sec	Will be high on a well performing system
Dir-DA	tracks not used per sec	Will be high on a well performing system
Disks	total SCSI command per sec	Used to see disk balance and plan growth
Disks	% util*	Used to see disk balance and plan growth
Devices	ios per sec	Can help identify the source application for any disk or DA imbalance
Devices	HA Kbytes transferred per sec	Can help identify the source application for any disk or DA imbalance
Devices	prefetched tracks per sec	Can help identify the source application of sequential activity

Note: the “% util” reported by EMC Performance Manager is not based on the theoretical maximum, but rather the point at which benchmark tests have shown to be the expected point where response times begin to increase dramatically with an increase in load. “% util” metrics can approach 100% before marked deterioration in performance is noted; however, the 100 % mark is an estimate, so may not be entirely accurate as it depends on the nature of the workload.

Familiarity with the values for the metrics noted above can give you a good understanding of “normal” operating conditions for the storage array. A properly constructed baseline that includes a measure of variability will help you to detect significant changes to these metrics before they become a problem, allowing you to take proactive measures. They also significantly help you to plan for growth.

EMC Performance Manager CLI

EMC Performance Manager provides a CLI that can be used to extract performance data from .btp files, and save it as a comma separated file (.csv). The CLI is named *pmcli.exe*, and is located in the root directory of Performance Manager. The EMC Performance Manager help files document use of the CLI.

The .csv output from pmcli.exe is the same as the .csv exports that can be done in the EMC Performance Manager GUI. However, the presence of the command line allows you to automate repetitive tasks with scripts and batch files.

For example, the command below creates a .csv containing selected performance data for array 9999 from the interval .btp collection taken on 20080201:

```
pmcli.exe -export -out C:\pmcli_test\sample\20080201.csv
-class symmetrix -id 9999 -c System -m "ios per sec","% writes","% hit","Kbytes read per sec","Kbytes written per sec" -type interval -date 20080201 –local
```

The .csv generated contains data such as the following (... denotes data that has been cut from the output for brevity):

EMC ControlCenter Performance Manager generated file from: X:\000190109999\interval\20080201.btp					
Data Collected for System					
Data Collected for System - 00019010999					
	02/01/2008 0:15	02/01/2008 0:30	…	02/01/2008 23:45	02/02/2008 0:00
ios per sec	9969.801	8251.143	…	4845.639	5014.774
% writes	17.692	14.566	…	19.008	17.768
% hit	79.387	81.165	…	61.418	63.727
Kbytes read per sec	325503.938	214421.047	…	359149.219	342096.063
Kbytes written per sec	213818.172	101971.672	…	85132.57	86028.414

Automation of Data Extracts

Because the output from pmcli.exe is predictable, you can choose to manipulate the data as you like. You can read data from the .csv files into a relational database, read data from the .csv files to generate web based reports, or simply use Microsoft Excel to view the data. The following section describes a relatively simple way to collate data to create a baseline for Symmetrix.

Step 1 - Extract performance data using pmcli.exe

The script below is a simple example of how pmcli.exe can be invoked from a windows shell script. This script uses pmcli.exe to create 30 .csv files from existing interval .btp collections. Note that your installation of WLA may not keep 30 days of interval .btp files.

```
'this script invokes pmcli.exe to export Symmetrix performance data.
Set WshShell = CreateObject("WScript.Shell")

'StartDate is the day/month/year of the .btp file we are starting from
StartDate = CDate("02/02/2008")

'loop to create a baseline with 30 days
For x = 0 to 29
```

```
        'DateAdd subtracts x days from the StartDate
        FileDate = DateAdd("d",-x,StartDate)
        BTPFile = Year(FileDate)

        if Month(FileDate) <10 then
                BTPFile = BTPFile&"0"&Month(FileDate)
        else
                BTPFile = BTPFile&Month(FileDate)
        end if
        if Day(FileDate) <10 then
                BTPFile = BTPFile&"0"&Day(FileDate)
        else
                BTPFile = BTPFile&Day(FileDate)
        end if

        'Path to the pmcli.exe
        Path = "C:\Program Files\EMC\PerformanceManager\pmcli.exe"
        'Flags for the command
        Flags = " -export -out C:\pmcli_test\sample\" & BTPFile &_
                ".csv -class symmetrix -id 9999 -c System -m " &_
                CHR(34) & "ios per sec" & CHR(34) & "," & CHR(34) &_
                "% writes" & CHR(34) & "," & CHR(34) & "% hit" & CHR(34) &_
                "," & CHR(34) & "Kbytes read per sec" & CHR(34) & "," &_
                CHR(34) & "Kbytes written per sec" & CHR(34) &_
                " -type interval -date "&BTPFile&" -local"

        'Execute the pmcli.exe
        WshShell.Run Path&Flags, 1, vbTrue

next

Set WshShell = nothing
```

Step 2 – Import the .csv files into an existing .xls

After exporting the performance data into a collection of .csv files, you can now import the .csv data into an .xls, to take advantage of Microsoft Excel's built in statistical functions and charting capabilities. After you have set up the .xls the first time with the charts and statistical analysis that you need, it is simple to import new data into the existing .xls to update the baseline, or create one for a different array, as below:

```
'this script copies data from CSV files to an existing workbook.
'The charts and error bars etc can be pre-defined, this script
'will just copy the data in.
Set objExcel = CreateObject("Excel.Application")
objExcel.Visible = True
objExcel.DisplayAlerts = FALSE

'StartDate is the month/day/year of the .btp file we are starting from
StartDate = CDate("03/02/2008")

'this opens an existing workbook
Set objWorkbookXLS = objExcel.Workbooks.Open("C:\pmcli_test\sample\Test.xls")

'loop to create a baseline with 30 days
For x = 0 to 29

        'FileDate counts days backwards, using DateAdd(interval,number,date)
        FileDate = DateAdd("d",-x,StartDate)
        CSVFile = Year(FileDate)

        if Month(FileDate) <10 then
                CSVFile = CSVFile&"0"&Month(FileDate)
        else
                CSVFile = CSVFile&Month(FileDate)
        end if
        if Day(FileDate) <10 then
                CSVFile = CSVFile&"0"&Day(FileDate)
        else
                CSVFile = CSVFile&Day(FileDate)
        end if
```

```
        'define the CSV
        str_open = "C:\pmcli_test\sample\disk\"& CSVFile & ".csv"
        'define the worksheet in the csv
        str_worksheetCSV = cstr(CSVFile)
        'open the CSV
        Set objWorkbookCSV = objExcel.Workbooks.Open(str_open)
        'set the worksheet in CSV
        Set objWorksheet = objWorkbookCSV.Worksheets(str_worksheetCSV)
        'set the range to copy.
'Range is set here for data with 15 minute interval
        Set objRangeCSV = objWorksheet.Range("A:A","CS:CS")
        objRangeCSV.Copy



        'Set to the original worksheet in the XLS.
        'Sheets are name DataSet0, DataSet1, etc
        Set objWorksheet = objWorkbookXLS.Worksheets("DataSet"&x)

        'paste to the XLS
        objWorksheet.paste

        'close the CSV
        objWorkbookCSV.close
next

'Save the XLS
objWorkbookXLS.SaveAs("C:\pmcli_test\sample\Test.xls")

objExcel.Quit
```

Step 3 – Summarize Data and Create Charts in the .xls

Some manual work is required to set up the .xls the first time you create a baseline or change metrics. However, the data from the .csv files all have the same format so it is easy to setup a worksheet that calculates the mean, standard deviation, and maximum values for all 30 days of data. As long as the same metrics are pulled in when the baseline is refreshed, and they are pulled into sheets in the .xls with the same names as before, the sheet that calculates the mean, standard deviation, and maximum values does not need to be changed each time a baseline is created. Similarly, charts created in Excel will also be refreshed with the new data.

Conclusion

As a supplement to the charts and graphs available within EMC Performance Manager, you can leverage the EMC Performance Manager CLI (pmcli.exe) to produce baseline performance data to aid in performance planning and analysis. Creating baselines is a laborious project if done manually, but some simple scripting can automate the movement of data from Performance Manager .btp files into Microsoft Excel spreadsheets and charts. If desired, the complexity of the automation can be increased to offer greater functionality, such as producing web based reports, loading the data into a relational database, or leveraging other commonly available tools, like Crystal Reports.