A 15x15 grid heatmap with a color scale from green (low) to red (high). The grid features several horizontal bands of color: a red band at row 5 (columns 1-5), a yellow band at row 9 (columns 4-9), and another red band at row 14 (columns 1-9). The text 'CLARiiON DIY' is centered in orange, and 'HeatMaps' is centered below it in a larger orange font. The version number 'v0.4.01' is located in the center of the grid.

CLARiiON DIY

HeatMaps

v0.4.01

Contents

- Purpose: 3
- Requirements:..... 3
 - PERL modules required by the script:..... 3
- Notes:..... 4
- Revision Notes: 5
- Usage: 7
- Databases..... 11
- Date Ranges 11
- Default Metrics: 12
- Debugging: 13
- Installation troubleshooting 14
- Examples:..... 15
 - Example 1 – Basic usage: 15
 - Example 2 – More usage:..... 16
 - Example 3 – Mash-ups:..... 17
 - Example 4 – Summary table: 18
 - Example 5 – Mouse overs / Tool-Tips..... 19
 - Example 6 – Highlights and Displaying Drive Types/Details 20
 - Example 7 – Retrieve NAR file & Date Stamp output 21
 - Example 8 – Displaying actual values and Backend IO stats..... 22
 - Example 9 – Displaying Detailed Data for an object 23
 - Example 10 – Changing min/mid/max colors 24
 - Example 11 – Steps 25
 - Example 12 – Retrieving NAR files and Database output 26
 - Example 13 – Generating Heatmaps from a database 27
 - Example 14 – Date Ranges..... 28

Purpose:

The heatmap script allows you to generate heatmaps of various metrics over time from NAR files generated from EMC CLARiiON/VNX arrays. It currently allows you to graph Disk, LUN, Storage Processor stats, and also generate Array wide, and per Bus graphs – these two latter sets are generated by aggregating Disk statistics.

The scripts generates the HeatMap graphs by taking a minimum and maximum for the attribute being graphed, breaking it down into 10 bands, averaging the interval into `--avg_interval <seconds>` (default 1800 seconds – 30 minutes) and mapping this against the 10 bands – the lowest band being green and the highest band being red.

Mashup graphs then take multiple attributes of the same type, and then average these to make a single graph.

Requirements:

- This script was developed and tested using Strawberry Perl (v5.12.3), but there is no reason it won't work with other flavours or versions of Perl, but it is possible that other Perl modules other than the one listed below may need to be installed
- It requires the Text::CSV module for Perl. Available here → <http://search.cpan.org/CPAN/authors/id/M/MA/MAKAMAKA/Text-CSV-1.21.tar.gz>, if you can get cpan working (run "cpan Text::CSV") it will be easy to install, otherwise you'll have to do it manually ...
 - Unzip the files into e.g. C:\Text-CSV-1.21 and run the following commands
 - C:\Text-CSV-1.21>perl Makefile.PL
 - C:\Text-CSV-1.21>dmake
 - C:\Text-CSV-1.21>dmake test
 - C:\Text-CSV-1.21>dmake install
- The script now uses SQLite as a temporary data store – this comes with the above version of Perl, again it is possible that the module may need to be downloaded and installed depending on the version of Perl installed.
- The script uses navisecli to
 1. Download the latest NAR file from the Storage Processor
 2. Query the array for its current configuration details.
 3. Convert the NAR files into CSV files and

PERL modules required by the script:

All of the following modules come with the strawberry distribution of Perl except the Text::CSV module

Text::CSV	used to parse CSV files
Getopt::Long	used to parse command-line options
Time::Piece	used to convert date/time to EPOCH and EPOCH to date/time
Time::Local	used to calculate GMT offset
DBI	used for SQLite connectivity
File::Which	used to find navisecli binary

If you experience problems with the script I would recommend making sure the above module are at the latest versions, if you are able to use cpan then this is fairly straight forward by running the following commands:

```
cpan Text::CSV
cpan Getopt::Long
cpan Time::Piece
cpan Time::Local
cpan DBI
cpan DBD::SQLite
cpan File::Which
```

Notes:

- The default min/max figures that I am using may be unrealistic; I am by no means an expert in analysing the performance of an array. Any feedback is welcome by posing to the relevant blog article found here <http://www.penguinpunk.net/blog/utilities/> or by e-mailing me directly at matharvest@gmail.com.
- The t_ attributes (total for the array and total per bus) are calculated by adding all the disk attributes for the array figures and all the disks attached to a bus for the bus figures and averaging them across the heatmap interval. These attributes are currently only displayed as actual figures rather than green -> red indicators and thus mash options will have no effect.

NOTE: These figures may differ from Storage Processor IOPS and MB/s stats due to RAID type inefficiencies and boundary crossings.

- Running the heatmap script solely across a NAR file will assume a single profile for all drives in the array, ie the default is to assume all drives are FC, this may produce unrealistic figures if there are a variety of drive types in the array ie SATA/FC/EFD or SAS/Nearline-SAS
- Querying an array against an older NAR file may also produce unexpected results, as drives may have been added or removed, and LUNs may have been destroyed or re-created with different IO profiles
- There are several output filename modifiers

--out <filename>	- The output filename, default is heatmap.html
--output_datetimestamp	- Add the current Date/Time stamp to the start of the filename (yyyymmdd-hhmmss)
--output_datestamp	- Add the current Date stamp to the start of the filename (yyyymmdd)
--output_arrayname	- Add the name of the array to the filename
--output_spname	- Add the name of the SP to the filename

The datatime stamp option will take precedence over the date stamp option.

The order that the modifiers will be applied is as follows [date_time | date] [sp_name] [array_name] [file_name]

- The LUN Bandwidth and IOPS attributes currently have an arbitrary min/max figure against them

IOPS total	10000
IOPS read & write	5000
Bandwidth total	500MB/s
Bandwidth read & write	250MB/s

The current code does not look at the underlying RAID type and number of spindles to try and calculate realistic maximum figures. The above maximums can be modified in code, or by using a configuration file.

- When just querying a NAR file (and not connecting to the array to query for extended disk details) drive that are configured as hot-spares (and not in use) or drives that are no bound will not return any performance data and will appear as blanks in the heatmap output
- The script is now configured to attempt to find the navisecli binary – if there are multiple instances of the binary installed on the system you need to be aware of the order in which they might be selected:
 1. Path/File Name specified in the command line
 2. Search the directories specified in the system environment variable PATH
 3. Search common installation locations
 - c:\Program Files\EMC\Navisphere CLI\NaviSECCLI.exe
 - c:\Program Files (x86)\EMC\Navisphere CLI\NaviSECCLI.exe
 - d:\Program Files\EMC\Navisphere CLI\NaviSECCLI.exe
 - d:\Program Files (x86)\EMC\Navisphere CLI\NaviSECCLI.exe
 - /opt/Navisphere/bin/navisecli

The first instance of a valid executable to be found will be the one that is selected and used.

Revision Notes:

0.3

- Change heatmap array in JavaScript to be comma separated array
- Remove style attributes from each cell and place in style definition

0.3.001

- Add more advanced command line processing

0.3.002

- Add command line options to allow for different attributes

0.3.003

- Add functionality to display multiple attributes

0.3.004

- Add ability to report on different object types disk, sp, etc.

0.3.005

- Clean-up report format
- Add command line options
- Add more SP metrics

0.3.006

- Add Summary option - display Max, Min and Avg for each metric type
- Add mash-up and mash-up only options
- Fix metric names

0.3.007

- Change heatmap array formatting

0.3.008

- Add options to allow naviseccli to query array for drive types

0.3.009

- Add drive details option
- Display drive Type / Size / RAID Group / Pool
- Allow drive groups to be selected and highlighted

0.3.010

- Config file

0.3.011

- Add LUN reporting
- Allow querying of specific types of LUNs
- Fix mashup calculations
- Stop displaying mashups alongside attributes when only one attribute is displayed
- Move control buttons to top of screen

0.3.012

- Add more tool-tips
- Fix bug with removed drives when displaying drive details

0.3.013

- Remove reliance on windows tool LogParser - replace functionality with SQLite
- Fixed bug where time was being converted to and from incorrect time zones
- Added more attributes to disk tool-tips

0.3.014

- Added disk types from VNX
- Added more disk sizes
- Fixed problem when using --display_drive_type and you had empty slots in the array
- Fixed problem with some LUN performance stats having an extra , in the first field (this is produced by naviseccli)
- Changed the way empty/unused trays or disks are displayed
 - If when not using the --get_drive_type option analyser does not return stats for a drive (i.e. hot-spares or unbound drives)
 - or when using the --get_drive_type option the drive state attribute is empty the drive will not be displayed

0.3.015

- Add ability to retrieve latest NAR file from the array
- Add command line option to add timestamp to output file
- Option --get_drive_type is now deprecated and is being replaced by the option --get_array_details

0.3.015.1

- Fix problem when using already downloaded NAR files (introduced by NAR download code)

0.3.016

- Add options to add array name and SP name to output file
- Fix --display_drive_type so that it displays empty drive slots as white, Removed / Failed drives as gray and unknown as green
- Add attributes to display total array and bus IOPS and Bandwidth
- Add --display_actual option to view actual IO stats
- Add read and write attributes for SP IOPS and bandwidth metrics
- Add --time_zone option
- Add the time zone to the heatmap output
- Add LUN bandwidth total, read & write and LUN IOPS total, read & write attributes
- Fix display problem when all of trays have their last disks configured as hotspares or not in use

0.3.017

- Change display options to allow controll of how many Disk, LUN and SP heatmaps per column
 - Add --disk_maps, --lun_maps and --sp_maps

- 0.3.018
 - Add --debug option to print detailed debug information
- 0.3.019
 - Search Path environment variable for naviseccli
 - Search common install locations for naviseccli
 - Improve cross browser support - tested on IE, Chrome and FireFox
 - Improve debug details - add module version reporting
 - Fix divide by zero bug in rendering routine
- 0.3.020
 - Add command line options to naviseccli when it is extracting data from the NAR file to enquote each CSV field. This should fix an issue that occurs when a LUN has multiple mount points
 - Several code style changes
 - Fix div by zero issue when using a config file and the d_w_iops attribute
- 0.3.021
 - Add command line options --min_color --mid_color --max_color to allow the user to select different color schemes
 - Add command line option --steps allow the user to control the number of steps or increments in reporting
 - Add command line option --detail_data when the user clicks on an object a graph of the objects utilization is shown
 - Add exit code checking after running naviseccli
 - Browser compatibility fixes
- 0.3.0211
 - Add --xxx_colour options
 - Remove case sensitivity for colours
 - Added FC SSD drive type
- 0.4
 - Add Database storage / retrieval for performance stats
 - Add command line option --output_db output the processed NAR file to the nominated database
 - Add command line option --input_db use the nominated database as the source of data for the heatmap
 - Change to use temporary tables for transitional data
- 0.4.0.1
 - Add command line option --s_date specify a start date/time must be in the format (with quotes if specifying date and time "mm/dd/yyyy hh:mm:ss")
 - Add command line option --e_date specify an end date/time
 - Add command line option --retrieve_all_nar when retrieving NAR files from the array, you can now retrieve all nar files (it won't overwrite files already downloaded)
 - Add command line option --process_only_new if you are downloading NAR files, only process files that haven't been downloaded previously
 - Add command line option --max_nar_files set the maximum number of files to download and process
 - Changed the way the script processes multiple NAR files, the script previously bunched all NAR files into a single naviseccli process, this was problematic if you were processing multiple large NAR files, the script now processes them one at a time

Usage:

The following are the command line options available for the script

`--navisecli <path\filename>`

These options allow you to set the location of the various 3rd party binaries that the script relies on

`--avg_interval <seconds>`

The interval that the analyser stats are averaged down to, ie 1800 seconds will set the average interval to 30 minutes

`--nar_file <filename>`

The input NAR file(s), more than one NAR file can be specified

`--time_zone <local|utc>`

Controls the way that times are displayed in the output, by default the script will display times in the local time zone.

`--retrieve_nar`

Retrieve the latest NAR file from the array and produce the heatmap

`--nar_location`

The location to store the NAR file once it has been retrieved from the array

`--output_datetimestamp`

Add date and time stamp prefix to the output filename in the format `yyyymmdd-hhmmss_<filename>`

`--output_datestamp`

Add date stamp prefix to the output filename in the format `yyyymmdd_<filename>`

`--output_arrayname`

Add the name of the array to the output filename

`--output_spname`

Add the name of the SP to the output filename

`--out <filename>`

The filename of the generated HTML output file, the default is HEATMAP.HTML

`--disk_maps`

Change the number of disk heatmaps displayed per column, the default is 1.

`--sp_maps`

Change the number of SP heatmaps displayed per column, the default is 10.

`--lun_maps`

Change the number of LUN heatmaps displayed per column, the default is 1.

`--min_color`

Change the low color on the heatmap the default is green

`--mid_color`

Change the mid color on the heatmap

`--max_color`

Change the high color on the heatmap the default is red

The colors that are currently available for use with the `min_color`, `mid_color` and `max_color` options are as follows `red,green,blue,yellow,cyan,magenta,purple,orange,black` and `white`

`--steps`

Change the granularity of the heatmap steps, for example on an attribute like % Utilization, if steps is set to 20, there will be different color bands for 0-4%, 5-9%, 10-14%,etc the default is 10 so color bands will be at 0-9%,10-19%,20-29%, etc

`--detail_data`

This option will allow you to display detail heat graph for a selected object.

--array_name <name>

The name of the array, this is for reporting purposes only

--display_actual

Gives the option to view actual IO stats for each element

--summary

Generate a summary for each attribute type, displaying minimum, average and maximum figures for each metric

Metric	Min	Max	Avg
Disk Utilization (%)	0.27	97.38	8.16
Disk Total Throughput (IO/s)	0.72	309.06	23.03
SP Utilization (%)	38.03	61.11	49.78
SP Total Throughput (IO/s)	5588.70	25693.24	10088.56

--mash

--mashonly

This generated a mash table for each metric type (currently only Storage Processor and Disk), averaging each of the metrics for each type down into a single table, allowing you to combine multiple metrics into a single table.

The --mash option will display the mash-up table alongside the selected metrics, and the --mashonly options will only display the mash-up tables.

--get_drive_type

--array_ip <ip address>

This option allows you to query the array (as it is currently configured) to determine drive types (currently only SATA II, FIBRE CHANNEL, and SATA II SSD). This options affects

- The IOPS calculations for drives.
- More details are supplied in mouse over / tool-tips.

NOTE: using this option may produce unexpected results if you are producing a heatmap from an old analyser file, as drives may have been added / removed, new LUNs bound / destroyed, and RAID groups / pools may have been created/extended or destroyed since the analyser file was produced.

--user_id <user id>

--pwd <password>

--scope <0|1>

These options are only used in conjunction with the --get_drive_type option, if you have cached credentials for the array configured then these options should not be necessary.

--display_drive_type

This option is only used in conjunction with the --get_drive_type option, it will display another drive table, and allow you to view the different drive type, drive size, pool and RAID Group layouts.

--disk_highlight

This option is only used in conjunction with the --get_drive_type and --display_drive_type options, it will highlight the corresponding disks in the other heatmaps, it will also allow you to select all of the drives that have the same attributes as displayed by the --display_drive_type option.

--config_file <filename>

--generate_config <filename>

These options allow you to generate a configuration file, using the --generate_config option, this will generate a config file with all of the default attributes, and to use a configuration file using the --config_file option.

--lun_type <type>

This option allows you to only display heatmap information about particular LUN types, multiple types can be specified

- all - Display stats for all LUNs
- lun - Display stats for all RAID group based LUNs
- meta - Display stats for all MetaLUNs
- private - Display stats for all Private LUNs - MetaLUN components and other reserved LUNs
- host - Display stats for all LUNs with host IO
- thin - Display stats for all thin LUNs
- pool - Display stats for all Pool LUNs

--debug <debug level>

Print extended debug information, the type of debug information is controlled by the debug level and valid values are from 0 thru 255

- 0 Informational
- 1 Basic
- 2 Perl Environment
- 4 CSV data
- 8 SQL insert commands
- 16 SQL query commands
- 32 SQL query data
- 64 SAN query data
- 128 Analyzer output

--help

This option will display the following help information.

Heatmap Generator

Usage: heatmap.4.0.1.pl <options>

Where options can be the following:

- naviseccli <path\filename> - The path to the naviseccli executable (C:\Program Files\EMC\Navisphere CLI\naviseccli.EXE)
- avg_interval <seconds> - The interval in seconds that the stats are averaged at (1800 seconds)
- nar_file <filename> - NAR input file(s) this option can be specified multiple times
- output_db - Output the processed NAR file to the nominated database
- input_db - Use the nominated database as the source of data for the heatmap
- s_date - Specify a start date/time must be in the format (with quotes if specifying date and time "mm/dd/yyyy hh:mm:ss")
- e_date - Specify an end date/time
- time_zone <local|utc> - Tells the script to report times in local or UTC time zone (default: local)
- retrieve_nar - Retrieve the latest NAR file from the array, and generate heatmap
- retrieve_all_nar - Retrieve all NAR file from the array
- process_only_new - If you are downloading NAR files, only process files that have not been downloaded previously
- max_nar_files - Set the maximum number of files to download and process
- nar_location - Specify the location to dump the latest NAR file retrieved from the array
- out <filename> - The output filename (heatmap.html)
- output_datetimestamp - Add the current Date/Time stamp to the start of the filename (yyyymmdd-hhmmss)
- output_datestamp - Add the current Date stamp to the start of the filename (yyyymmdd)
- output_arrayname - Add the name of the array to the filename
- output_spname - Add the name of the SP to the filename
- disk_maps <number> - Change the number of Disk maps per column (default=1)
- sp_maps <number> - Change the number of SP maps per column (default=10)
- lun_maps <number> - Change the number of LUN maps per column (default=1)
- min_color - Change the low color (default=green)
- mid_color - Change the mid color
- max_color - Change the high color (default=blue)
red,green,blue,yellow,cyan,magenta,purple,orange,black,white
- steps - Change the granularity of the stats (default=10)
- detail_data - Display detail graph for a selected object
- array_name <name> - Set the name of the array in the report
- summary - Displays a summary of each metric
- mash - Creates a mash-up of each metric per object type, and displays alongside other metrics
- mashonly - Creates a mash-up of each metric per object type, and only displays the mash-ups
- get_drive_type - <deprecated> see --get_array_details
- get_array_details - Query Array to get drive type information
- array_ip <ip address> - Set the IP address of the array
- user_id <user id> - Set the user ID to log into the array
- pwd <password> - Set the user password of the array
- scope <0|1> - Set the Scope of the array account
- display_drive_type - Display the drive types in the charts
- disk_highlight - Highlight drives on mouse over
- display_actual - Allows the user to view performance figures
- config_file <filename> - Use a configuration file to set attribute min/maxes

```

--generate_config <filename> - Generate a configuration file using the defined defaults

--debug <level>                - Prints debug information
--help                          - This help

--lun_type <type>              - Which LUN types to query, multiple types can be specified
    all                         - Display stats for all LUNs
    lun                         - Display stats for all RAID group based LUNs
    meta                        - Display stats for all MetaLUNs
    private                     - Display stats for all Private LUNs - MetaLUN components and other
                                reserved LUNs
    host                        - Display stats for all LUNs with host IO
    thin                        - Display stats for all thin LUNs
    pool                        - Display stats for all Pool LUNs

--attrib <attribute>          - Set the attribute to graph, where attribute can be the following
    d_utilization              - Display stats based on disk utilization (%)
    d_iops                     - Display stats based on disk Total Throughput (IOPS)
    d_r_iops                   - Display stats based on disk read IOPS
    d_w_iops                   - Display stats based on disk write IOPS
    d_queue                    - Display stats based on disk Queue Length
    d_b_queue                  - Display stats based on disk Average Busy Queue Length
    d_response                 - Display stats based on disk Response Time
    d_service                  - Display stats based on disk Service Time
    d_bandwidth                - Display stats based on disk Total Bandwidth
    d_r_bandwidth              - Display stats based on disk Read Bandwidth
    d_w_bandwidth              - Display stats based on disk Write Bandwidth
    d_r_size                   - Display stats based on disk Read Size
    d_w_size                   - Display stats based on disk Write Size
    d_seek                     - Display stats based on disk Average Seek Distance

    s_utilization              - Display stats based on SP utilization
    s_response                 - Display stats based on SP Response Time
    s_bandwidth                - Display stats based on SP Total Bandwidth
    s_r_bandwidth              - Display stats based on SP read Bandwidth
    s_w_bandwidth              - Display stats based on SP write Bandwidth
    s_iops                     - Display stats based on SP Total Throughput (IOPS)
    s_r_iops                   - Display stats based on SP read Throughput (IOPS)
    s_w_iops                   - Display stats based on SP write Throughput (IOPS)
    s_queue                    - Display stats based on SP Queue Length
    s_b_queue                  - Display stats based on SP Average Busy Queue Length
    s_service                  - Display stats based on SP Service Time
    s_c_dirty                  - Display stats based on SP Cache Dirty Pages (%)
    s_c_flush                  - Display stats based on SP Cache Flush Ratio
    s_c_flush_mb               - Display stats based on SP Cache MBs Flushed (MB/s)
    s_c_hw_flush               - Display stats based on SP Cache High Water Flush On
    s_c_i_flush                - Display stats based on SP Cache Idle Flush On
    s_c_lw_flush               - Display stats based on SP Cache Low Water Flush Off
    s_wc_flush                 - Display stats based on SP Write Cache Flushes/s
    s_fc_dirty                 - Display stats based on FAST Cache Dirty Pages (%)
    s_fc_flush_mb              - Display stats based on FAST Cache MBs Flushed (MB/s)

    l_utilization              - Display stats based on LUN utilization (%)
    l_queue                    - Display stats based on LUN Queue Length
    l_b_queue                  - Display stats based on LUN Average Busy Queue Length
    l_response                 - Display stats based on LUN Response Time
    l_service                  - Display stats based on LUN Service Time
    l_bandwidth                - Display stats based on LUN Total Bandwidth
    l_r_bandwidth              - Display stats based on LUN read Bandwidth
    l_w_bandwidth              - Display stats based on LUN write Bandwidth
    l_iops                     - Display stats based on LUN Total Throughput (IOPS)
    l_r_iops                   - Display stats based on LUN read Throughput (IOPS)
    l_w_iops                   - Display stats based on LUN write Throughput (IOPS)

```

The following option show a consolidated array/bus view averaged over 1800 second intervals

```

t_a_iops                       - Display total array IOPS
t_a_r_iops                     - Display total array read IOPS
t_a_w_iops                     - Display total array write IOPS
t_a_bandwidth                  - Display total array bandwidth (MB/s)
t_a_r_bandwidth                - Display total array read bandwidth (MB/s)
t_a_w_bandwidth                - Display total array write bandwidth (MB/s)
t_b_iops                       - Display bus IOPS averaged over
t_b_bandwidth                  - Display bus bandwidth (MB/s)
t_b_r_iops                     - Display bus read IOPS
t_b_w_iops                     - Display bus write IOPS
t_b_r_bandwidth                - Display bus read bandwidth (MB/s)
t_b_w_bandwidth                - Display bus write bandwidth (MB/s)

```

Running the script without any --nar_file options will result in the script prompting the user to supply NAR file(s)

Databases

You can now process the NAR files into a database, to allow you to graph historical data against current data more easily. In the past you would have to re-process the NAR files each time you wanted to generate a new graph. This process now allows you to process a NAR file once, and then query the database multiple times and add more data to the database

NOTE: you should use a separate database for each arrays NAR files that you process, mixing arrays in a single database will produce some unexpected results

The database size will depend on the size of the NAR files (and the complexity of the configuration of your array), however it appears that the database will be a little over 2 x the size of each of the NAR files that you ingest, and this is partly due to the fact that all metrics are stored in the database.

Date Ranges

The options `--s_date` and `--e_date` allow you to specify start and end dates to process. Please note that the format for these dates is currently very strict

`DD/MM/YYYY HH:MM:SS` **Note the 4 digit year**

If you specifying date and time then you will need to enclose the date/time string in quotes

If the script has problems parsing the string it will fail with the error

```
"Error parsing time at C:/strawberry/perl/lib/Time/Piece.pm line 469."
```

Default Metrics:

Below is a list of the default minimum and maximum metrics defined in the script, these figures may not be realistic for your array, and may need modification to suite your array, a configuration file can be generated with the defaults using the --generate_config <filename> script option, and the file can then be used using the --config_file <filename> option

Attribute	LUN	Disk	SP
	min/max	min/max	min/max
Utilization (%)	0/100	0/100	0/100
Response Time (ms)	0/20	0/100	0/50
Bandwidth	0/500	0/50	0/1000
Read Bandwidth	0/250	0/50	0/1000
Write Bandwidth	0/250	0/50	0/1000
IOPS	0/10000	0/360	0/30000
Read IOPS	0/5000	0/360	0/30000
Write IOPS	0/5000	0/360	0/30000
Queue Length	0/100	0/100	0/800
Average Busy Queue Length	0/100	0/100	0/800
Service Time (ms)	0/20	0/100	0/50
Cache Dirty Pages (%)			0/100
Cache Flush Ratio			0/1
Cache MB Flush (MB/s)			0/500
Cache High Water Flush On			0/1000
Cache Idle Flush On			0/1000
Cache Low Water Flush Off			0/1000
Write Cache Flushes/s			0/5000
FAST Cache Dirty Pages (%)			0/100
FAST Cache MBs Flushed (MB/s)			0/500
Read Size		0/512	
Write Size		0/512	
Average Seek Distance		0/500	
IOPS (Fibre Channel) *		0/360	
IOPS (SATA II) *		0/150	
IOPS (Solid State) *		0/5000	
IOPS (Nearline SAS) *		0/150	
IOPS (SAS) *		0/360	
IOPS (SATA II Solid State) *		0/5000	

* these options are only used when array discovery is performed, when not used a default IOPS figure will be used

Debugging:

Using the --debug <debug level> option will print extended information to std_out, along with other general informational output, the debug values can be added together to produce a mixed output.

Bit	8	7	6	5	4	3	2	1	0
Debug Value	128	64	32	16	8	4	2	1	0
Debug Data Type	Analyser Output								
	SAN Query Data								
	SQL Query Data								
	SQL Query Commands								
	SQL Insert Commands								
	CSV Data								
	Perl Environment								
	Basic Debug								
	Information								

Setting the debug level to 3 will show the Perl Environment, Basic Debug and informational debug data types

NOTE: using the debug options to print CSV Data (4), SQL Insert Commands(8), SQL Query Data(32) and SQL Query Data(64) can produce very large amounts of data, especially if querying multiple attribute types.

For most debugging requirements debug level 3 should be sufficient.

Installation troubleshooting

1. Some perl installations don't come with all of the required modules bundled, or come with older versions of the modules. Error messages such as the following:

```
Can't locate Text/CSV.pm in @INC ...
install_driver(SQLite) failed: Can't locate DBD/SQLite.pm in @INC ...
```

Indicate that a module has not been installed, in the first case Text::CSV and the second DBD::SQLite

2. Error messages like the following

```
Can't locate object method "epoch" via package...
```

Might indicate the module Time::Piece is an older version, and needs to be updated. The module Time::Piece version - 1.15_05 will produce the above error.

You can run the heatmap script in debug level 2 to find out the currently installed versions of the required modules

The following modules and their versions are known to work correctly with the script

```
Text::CSV           : 1.21
Getopt::Long        : 2.38
Time::Piece         : 1.20
Time::Local         : 1.1901_01
DBI                 : 1.616
DBD::SQLite         : 1.33
File::Which         : 1.09
```

3. Error messages like the following

```
Not a valid archive file
```

Might indicate an older/unsupported version of navisecli installed on the client that you are running the script from.

You should ensure that you are running the latest supported version of navisecli that will run with the version of FLARE code that you are running on your array.

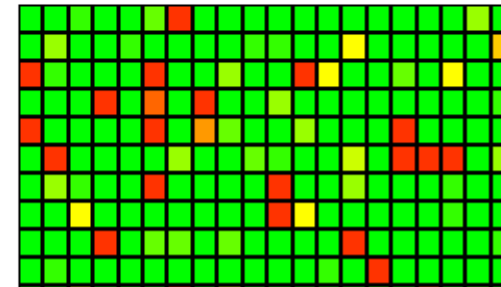
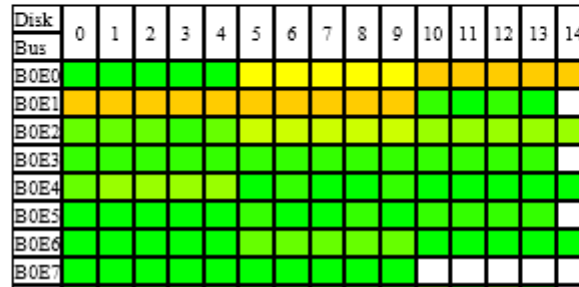
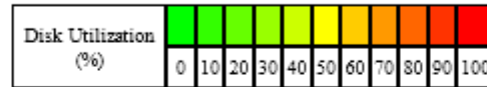
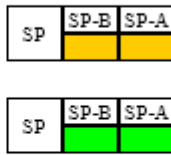
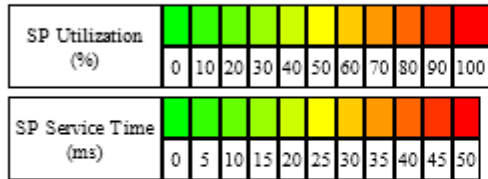
Example 2 - More usage:

This will produce a basic heatmap of multiple metrics, note: different metrics for the same object type can be displayed

```
perl heatmap.3.021.pl --nar_file nar_archive\test.nar --attrib d_utilization --attrib s_utilization --attrib s_service --attrib l_utilization
Adding file: nar_archive\test.nar
Processing NAR File:nar_archive\test.nar
.....
Processing NAR File:nar_archive\test.nar
.....
Processing NAR File:nar_archive\test.nar
.....
Querying CSV file for Disk Utilization (%)
Querying CSV file for SP Utilization (%)
Querying CSV file for SP Service Time (ms)
Querying CSV file for LUN Utilization (%)
Processing CSV file - tmp_heatmap_d_utilization_d.csv
Processing CSV file - tmp_heatmap_s_utilization_s.csv
Processing CSV file - tmp_heatmap_s_service_s.csv
Processing CSV file - tmp_heatmap_l_utilization_l.csv
Generating heatmap file: heatmap.html
```

Source File nar_archive\test.nar Time 2011-10-22 16:00:00

Source Array Delay (ms):



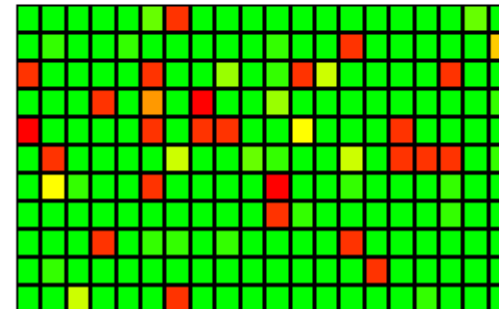
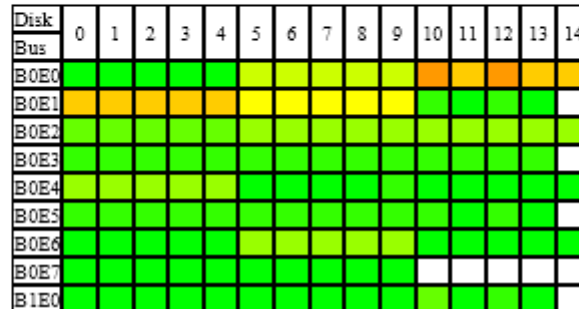
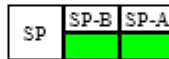
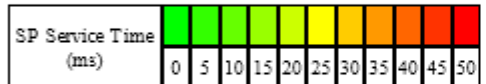
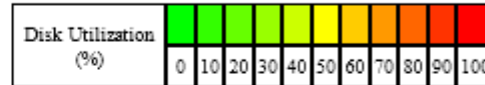
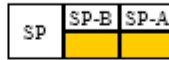
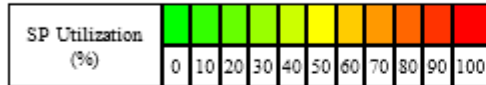
Example 4 – Summary table:

This will produce a summary of each metric displayed, showing the min/max and average

```
perl heatmap.3.021.pl --nar_file nar_archive\test.nar --attrib d_utilization --attrib l_utilization --attrib s_utilization --attrib s_service --summary
Adding file: nar_archive\test.nar
Processing NAR File:nar_archive\test.nar
.....
Processing NAR File:nar_archive\test.nar
.....
Processing NAR File:nar_archive\test.nar
.....
Querying CSV file for Disk Utilization (%)
Querying CSV file for LUN Utilization (%)
Querying CSV file for SP Utilization (%)
Querying CSV file for SP Service Time (ms)
Processing CSV file - tmp_heatmap_d_utilization_d.csv
Processing CSV file - tmp_heatmap_l_utilization_l.csv
Processing CSV file - tmp_heatmap_s_utilization_s.csv
Processing CSV file - tmp_heatmap_s_service_s.csv
Generating heatmap file: heatmap.html
```

Source File nar_archive\test.nar Time 2011-10-22 16:30:00

Source Array Delay (ms):



Metric	Min	Max	Avg
Disk Utilization (%)	0.27	97.38	8.16
LUN Utilization (%)	0.00	100.00	3.69
SP Utilization (%)	38.03	61.11	49.78
SP Service Time (ms)	0.02	0.09	0.05

Example 6 – Highlights and Displaying Drive Types/Details

This option will display a table that will show drive type (ie FC/SATA/SAS,etc) drive sizes, RAID groups and pools, if you select a particular set – ie click on the 450GB disks, all of these drives will be highlighted, across all disk heatmaps. With the --disk_highlight option, hovering the mouse over a particular disk will highlight that disk in other heatmaps

```
perl heatmap.3.021.pl --nar_file nar_archive\test.nar --attrib d_utilization --attrib l_utilization --attrib s_utilization --attrib s_service --summary --get_drive_type --array_ip 1.1.1.1 --user_id my_user_id --scope 0 --pwd my_password --disk_highlight --display_drive_type
```

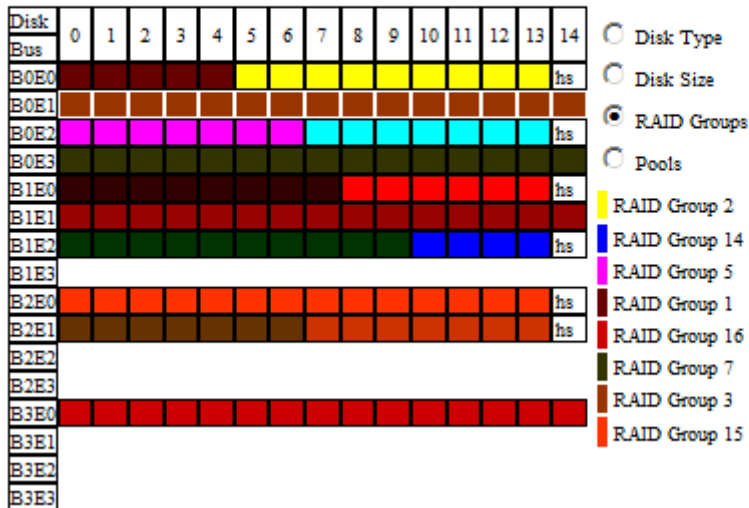
```
Querying array for LUN characteristics
Querying array for Pool LUN characteristics
Querying array for drive characteristics
Querying array for pool characteristics
Adding file: nar_archive\test.nar
Processing NAR File:nar_archive\test.nar
```

```
Processing NAR File:nar_archive\test.nar
```

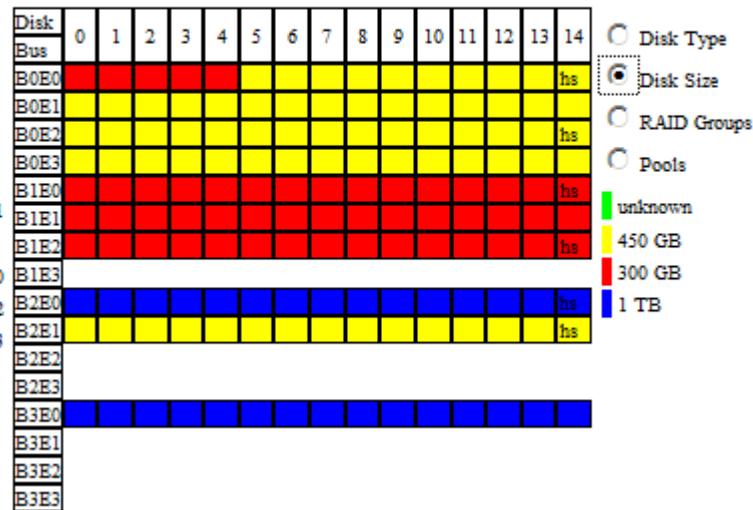
```
Processing NAR File:nar_archive\test.nar
```

```
Processing CSV file - tmp_analyzer_heatmap_l.csv
Processing CSV file - tmp_analyzer_heatmap_d.csv
Processing CSV file - tmp_analyzer_heatmap_s.csv
Processing Disk Attribute Disk Utilization (%)
Processing LUN Attribute LUN Utilization (%)
Processing SP Attribute SP Utilization (%)
Processing SP Attribute SP Service Time (ms)
Generating heatmap file: heatmap.html
```

Drive Details



Drive Details



Example 7 – Retrieve NAR file & Date Stamp output

These options allow you to retrieve the NAR file from the array and produce a heatmap in one pass

```
perl heatmap.3.021.pl --retrieve_nar --nar_location nar_archive\ --array_ip 1.1.1.1 --user_id my_user_id --scope 0 --pwd my_password --attrib d_utilization --get_array_details --output_datestamp --attrib s_utilization
Retrieving NAR file CK000000000000_SPB_2012-03-02_14-20-34-GMT_P10-00.nar Dated 03/03/2012 00:24:03 saving file to nar_archive\
Downloaded NAR file successfully
Querying array for LUN characteristics
Querying array for Pool LUN characteristics
Querying array for drive characteristics
Querying array for pool characteristics
Adding file: nar_archive\CK000000000000_SPB_2012-03-02_14-20-34-GMT_P10-00.nar
Processing NAR File:nar_archive\CK000000000000_SPB_2012-03-02_14-20-34-GMT_P10-00.nar
....
Processing NAR File:nar_archive\CK000000000000_SPB_2012-03-02_14-20-34-GMT_P10-00.nar
....
Processing CSV file - tmp_analyzer_heatmap_d.csv
Processing CSV file - tmp_analyzer_heatmap_s.csv
Processing Disk Attribute Disk Utilization (%)
Processing SP Attribute SP Utilization (%)
Generating heatmap file: 20120303_heatmap.html
```

Example 8 - Displaying actual values and Backend IO stats

```
perl heatmap.3.021.pl --retrieve_nar --nar_location nar_archive --get_array_details --array_ip 1.1.1.1 --user_id my_user_id --pwd my_password --scope 0 --attrib d_iops -
-display_actual --attrib d_utilization --attrib s_iops --attrib s_utilization --attrib s_bandwidth --attrib t_a_iops --attrib t_a_bandwidth --attrib t_b_iops --attrib
t_b_bandwidth --summary
```

Retrieving NAR file CK000000000000_SPA_2012-03-12_08-04-17-GMT_P10-00.nar Dated 03/12/2012 18:06:35 saving file to nar_archive/

NAR file already exists - continuing without download

Querying array name

Querying SP name

Querying array characteristics

Querying array for LUN characteristics

Querying array for Pool LUN characteristics

Querying array for drive characteristics

Querying array for pool characteristics

Adding file: nar_archive/CK000000000000_SPA_2012-03-12_08-04-17-GMT_P10-00.nar

Processing NAR File:nar_archive/CK000000000000_SPA_2012-03-12_08-04-17-GMT_P10-00.nar

....

Processing NAR File:nar_archive/CK000000000000_SPA_2012-03-12_08-04-17-GMT_P10-00.nar

....

Processing CSV file - tmp_analyzer_heatmap_d.csv

Processing CSV file - tmp_analyzer_heatmap_s.csv

Processing Total Attribute Total Array IOPS

Processing Total Attribute Total Array Bandwidth (MB/s)

Processing Total Attribute Total Bus IOPS

Processing Total Attribute Total Bus Bandwidth (MB/s)

Processing Disk Attribute Disk Total Throughput (IO/s)

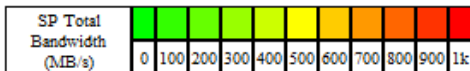
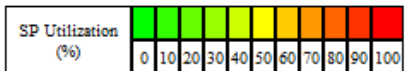
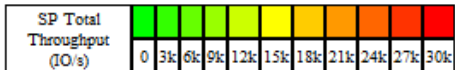
Processing Disk Attribute Disk Utilization (%)

Processing SP Attribute SP Total Throughput (IO/s)

Processing SP Attribute SP Utilization (%)

Processing SP Attribute SP Total Bandwidth (MB/s)

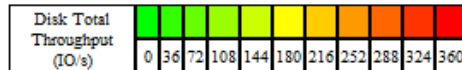
Generating heatmap file: heatmap.html



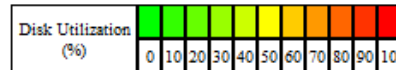
SP	SP-B	SP-A
	9667	8613

SP	SP-B	SP-A
	61	48

SP	SP-B	SP-A
	232	205



Disk	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Bus	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
BOE0	4	3	2	0	0	36	36	35	34	35	22	22	22	23	22
BOE1	17	16	17	17	17	16	16	16	16	0	0	0	0	0	0
BOE2	121	126	126	126	121	19	20	21	21	19	16	15	14	15	15
BOE3	74	72	73	72	72	71	71	70	70	70	70	70	70	0	0
BOE4	38	38	38	38	38	15	14	16	15	15	44	44	44	44	44
BOE5	7	7	7	7	7	4	4	4	4	4	1	1	1	1	0
BOE6	1	1	1	1	0	13	13	13	13	13	6	6	6	6	6



Disk	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Bus	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
BOE0	1	1	1	1	1	10	10	10	10	10	7	7	7	7	7
BOE1	8	8	8	7	8	7	6	6	7	7	0	0	0	0	0
BOE2	29	28	28	29	29	8	8	8	8	8	8	8	7	7	7
BOE3	25	25	25	24	24	24	24	23	22	22	23	22	23	23	0
BOE4	17	17	16	17	17	14	14	15	14	15	22	22	21	21	22
BOE5	2	2	2	2	2	1	1	1	1	1	0	0	1	0	0
BOE6	0	0	0	0	0	5	4	5	5	5	1	1	1	1	1

Total Array IOPS	Array
	23490

Total Array Bandwidth (MB/s)	Array
	1012

Total Bus IOPS	0	1	2	3	4	5	6	7
	3189	3490	3798	3639	2210	3601	2326	1234

Total Bus Bandwidth (MB/s)	0	1	2	3	4	5	6	7
	136	118	143	216	78	156	98	63

Metric	Min	Max	Avg
Total Array	11352	28755	21805

Example 9 - Displaying Detailed Data for an object

The `--detail_data` option allows you when selecting (clicking) an object on the heatmap to display the data over time for that object

```
perl heatmap.3.021.pl --nar_file CK000000000000_SPA_2012-06-05_20-00-50-GMT_P10-00.nar --attrib s_iops --detail_data
```

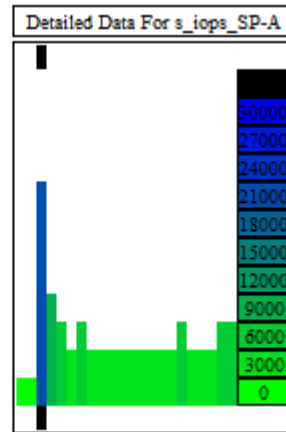
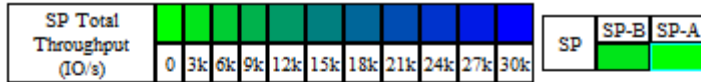
```
Adding file: CK000000000000_SPA_2012-06-05_20-00-50-GMT_P10-00.nar
```

```
Processing NAR File: CK000000000000_SPA_2012-06-05_20-00-50-GMT_P10-00.nar
```

```
Processing CSV file - tmp_analyzer_heatmap_s.csv
```

```
Processing SP Attribute SP Total Throughput (IO/s)
```

```
Generating heatmap file: heatmap.html
```



Example 12 – Retrieving NAR files and Database output

In this example we are using the retrieve_all_nar option, with a maximum number of files to process set at 2, you will see that the first file encountered is skipped because it already exists. Running the script this way will result in the NAR file(s) being processed into a SQLite database (named somedb8.db)

```
perl heatmap.4.0.1.pl --array_ip 1.1.1.1 --user_id my_userid --pwd my_password --scope 0 --output_db somedb8.db --retrieve_all_nar --process_only_new --max_nar_files 2
Retrieving NAR file CK200080600679_SPA_2013-03-31_20-00-49-GMT_P10-00.nar Dated saving file to ./
NAR file already exists - continuing without download
Retrieving NAR file CK200080600679_SPA_2013-04-01_20-00-50-GMT_P10-00.nar Dated saving file to ./
Downloaded NAR file successfully
Retrieving NAR file CK200080600679_SPA_2013-04-02_20-00-49-GMT_P10-00.nar Dated saving file to ./
Downloaded NAR file successfully
Processing NAR File:./CK200080600679_SPA_2013-04-01_20-00-50-GMT_P10-00.nar for object type d
Processing NAR File:./CK200080600679_SPA_2013-04-01_20-00-50-GMT_P10-00.nar for object type al
Processing NAR File:./CK200080600679_SPA_2013-04-01_20-00-50-GMT_P10-00.nar for object type s
Processing CSV file - tmp_analyzer_heatmap_d.csv
Processing CSV file - tmp_analyzer_heatmap_al.csv
Processing CSV file - tmp_analyzer_heatmap_s.csv
Processing NAR File:./CK200080600679_SPA_2013-04-02_20-00-49-GMT_P10-00.nar for object type d
Processing NAR File:./CK200080600679_SPA_2013-04-02_20-00-49-GMT_P10-00.nar for object type al
Processing NAR File:./CK200080600679_SPA_2013-04-02_20-00-49-GMT_P10-00.nar for object type s
Processing CSV file - tmp_analyzer_heatmap_d.csv
Processing CSV file - tmp_analyzer_heatmap_al.csv
Processing CSV file - tmp_analyzer_heatmap_s.csv
```

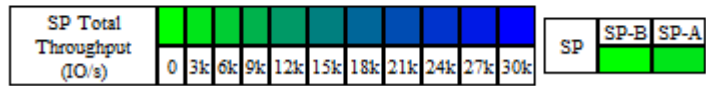
Example 13 - Generating Heatmaps from a database

The following example will generate a heatmap from a SQLite database (generated in the above example)

```
perl heatmap.4.0.1.pl --attrib s_iops --input_db somedb8.db  
Source Database: somedb8.db  
Processing SP Attribute SP Total Throughput (IO/s)  
Generating heatmap file: heatmap.html
```

Source File Time Sat Mar 16 10:30:00 2013 GMT +10

Source Array << >> Delay (ms):



Example 14 - Date Ranges

This example demonstrates how to use the date range options

```
perl heatmap.4.0.1.pl --attrib s_iops --input_db somedb8.db --s_date "18/3/2013" --e_date "19/3/2013 12:00:00"  
Source Database: somedb8.db  
Processing SP Attribute SP Total Throughput (IO/s)  
Generating heatmap file: heatmap.html
```