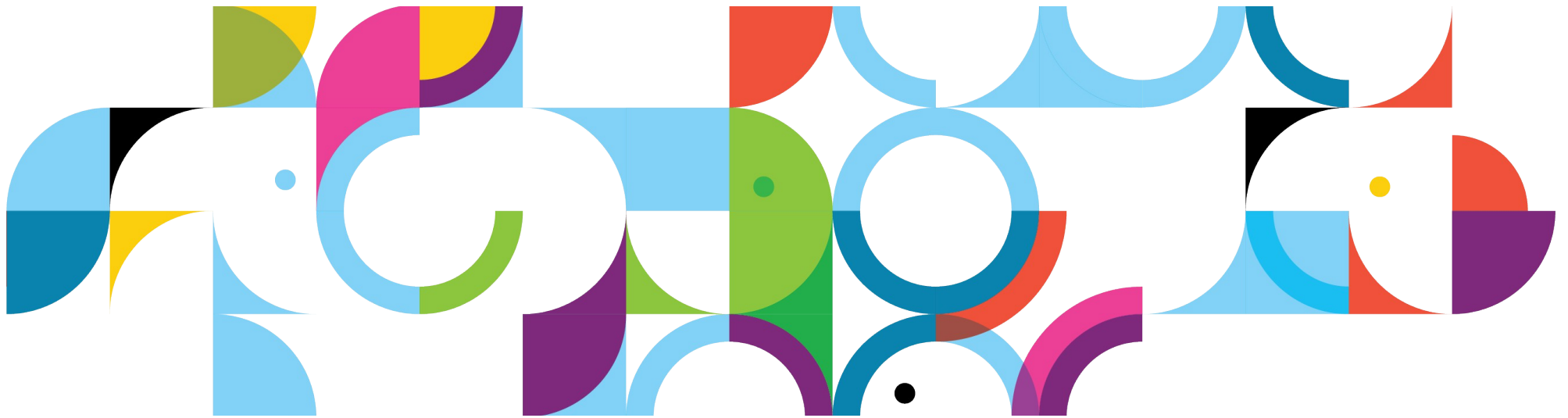


# NMONVisualizer

Processing & Analyzing nmon Graphically



nmon is a tool for viewing & collection system data

## AIX

- Built in as part of topas
- Closed source

## Linux

- Open source
- Download and install from <http://nmon.sourceforge.net/pmwiki.php?n=Site.Download>
  - Make sure you get the right one for your distribution
-

# nmon has a graphical mode

Just run from the command line.

- nmon
- Then hit letters to show data
  - c(cpu), d(isk), n(etwork) is a good start but it may not fit on the screen
  - h(elp) for more
  - q(uit) to exit

**This is live data.  
It does not write to a file.**

```
nmon-14g-----Hostname=chaperone-----Refresh= 2secs -----16:18.25-----
-----
# # # # ##### # #
## # ## ## # ## #
# # # # ## # # # # #
# ## # # # # # # #
# # # # # # # # #
# # # # ##### # #
-----
For help type H or ...
nmon -? - hint
nmon -h - full

To start the same way every time
set the NMON ksh variable

Use these keys to toggle statistics on/off:
c = CPU          l = CPU Long-term    - = Faster screen updates
m = Memory       j = Filesystems        + = Slower screen updates
d = Disks        n = Network           V = Virtual Memory
r = Resource    N = NFS               v = Verbose hints
k = kernel      t = Top-processes    . = only busy disks/procs
h = more options                               q = Quit
```

```
nmon-14g-----[H for help]-----Hostname=chaperone-----Refresh= 2secs -----16:20.59-----
CPU Utilisation
-----
CPU  User%  Sys%  Wait%  Idle|0      |25      |50      |75      |100|
1   0.5    1.0   0.0    98.5|>
2   1.0    3.5   0.0    95.5|s>
3   0.5    0.5   0.0    99.0|>
4   0.0    0.0   0.0    100.0|>
5   1.5    4.0   0.0    94.4|ss>
6   0.5    1.5   0.0    98.0|>
7   1.5    4.6   0.0    93.8|ss>
8   1.0    1.5   0.0    97.5|>
9   1.0    0.5   0.0    98.5|>
10  0.5    0.5   0.0    99.0|>
11  0.0    0.0   0.0    100.0|>
12  0.5    2.0   0.0    97.5|s>
13  1.0    4.0   0.0    94.9|ss>
14  1.0    6.1   0.0    92.9|sss>
15  0.0    0.5   0.0    99.5|>
16  1.5    0.5   0.0    98.0|>
17  2.5    4.0   0.0    93.5|tss>
18  0.5    0.0   0.0    99.5|>
19  1.0    3.0   0.0    96.0|s>
20  1.5    4.1   0.0    94.4|ss>
21  0.5    0.5   0.0    99.0|>
22  0.0    0.0   0.0    100.0|>
23  0.0    0.5   0.0    99.5|>
24  0.5    0.5   0.0    99.0|>
-----
Avg  0.8    1.8   0.0    97.4|>
```

# Write nmon data to a file

## Command line switch

- -f ⇒ create a file named <hostname>\_<yyMMdd>\_<HHmm>.nmon in the current directory
- -F ⇒ specify the file name

## Also need to specify

- -c – number of samples, default 3000
- -s – time between samples in seconds, default 30
- Samples \* count = time nmon will run

## If you need to stop nmon, do not just kill it!

- Use user signal 2 to have nmon stop after writing the next record
  - Prevents corruption of the last sample
- Linux – kill -s USR2 <pid>
- AIX – kill -s SIGUSR2 <pid>

# I have nmon Data, now what?

## NMONAnalyzer

- Excel spreadsheet macro
  - Do you have Excel?
  - Slower
- 
- Only a single file at a time
- 
- Time ranges only specified in terms of samples

## NMONVisualizer

- Java GUI (Swing)
  - Runs anywhere
  - Faster
- 
- Multiple files at once
  - Either from a single server or multiple servers
- 
- Absolute and relative time ranges
  - Handles time zones too
- 
- Also processes IOStat, Verbose GC & ESXTop files

## Getting NMONVisualizer

**NMONVisualizer can be downloaded from the link on <http://nmonvisualizer.github.io/nmonvisualizer/>**

**NMONVisualizer is an open source project**

- GitHub Project Page – <https://github.com/nmonvisualizer/nmonvisualizer>
- GitHub Releases – <https://github.com/nmonvisualizer/nmonvisualizer/releases>

# Running NMONVisualizer

## Windows & Mac

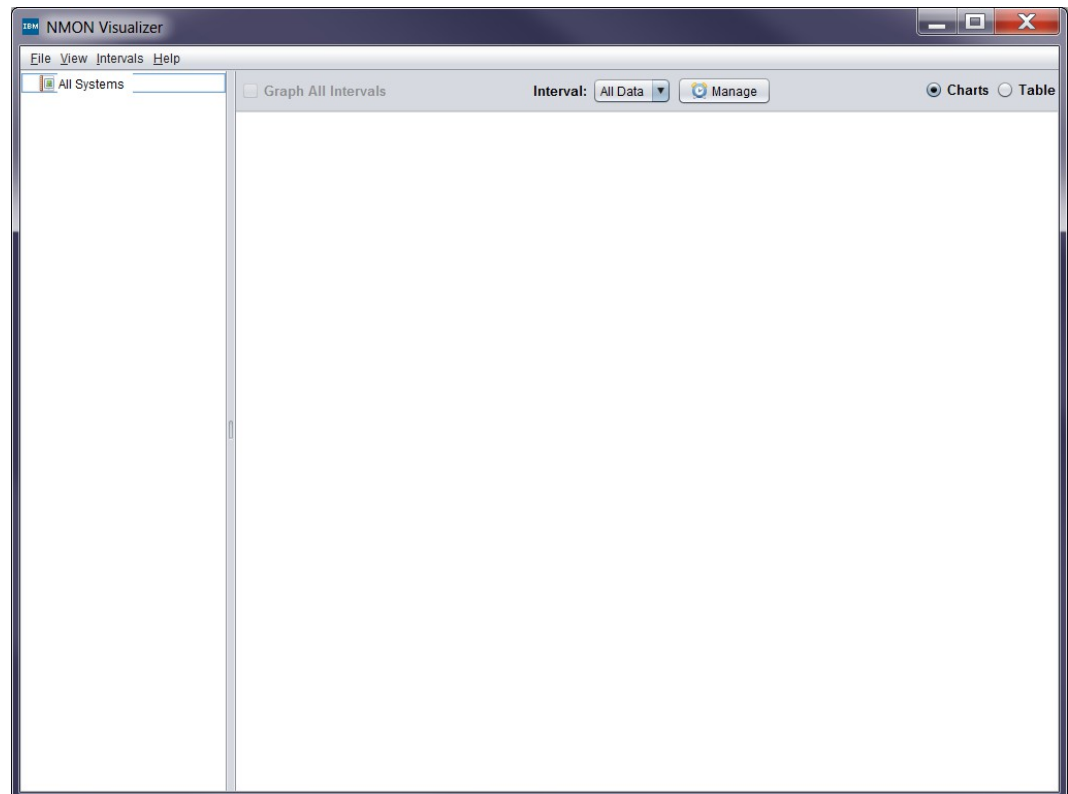
- Can just double click if a JRE is installed

## Linux or AIX

- Command line (`java -jar`)
- XWindows works too

## Make sure you have max heap (-Xmx) set high enough

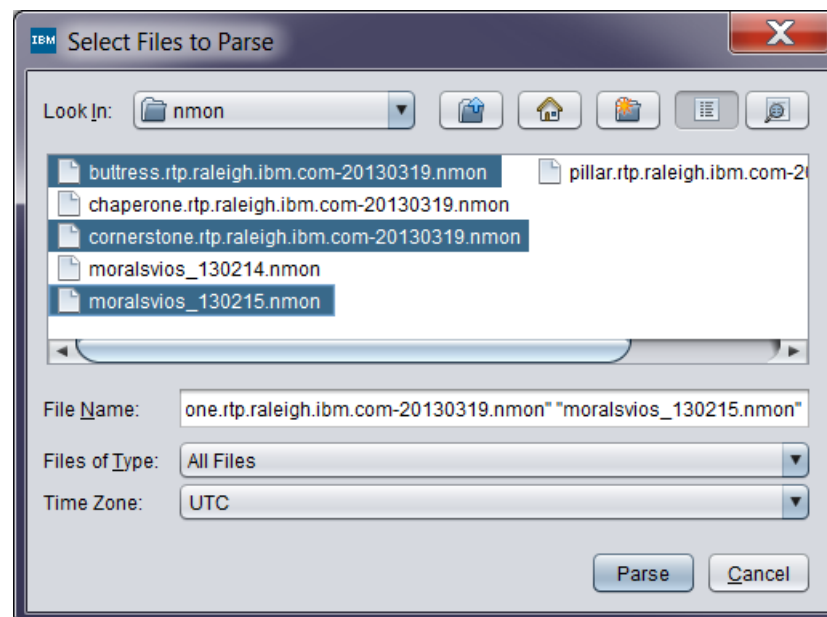
- May need 1 or 2 GB for a 'large number' of files
  - Depends on the amount of data per file



## Parsing nmon data

### Multiple ways to get files into the program

- File → Load or CTRL-O
  - Opens file dialog
    - Multi-selection supported
    - Directories supported; fully recursive
- Drag and Drop
  - Drag from file viewer (Explorer, Finder, etc) into left hand pane of UI





# NMONVisualizer displays reports at multiple levels

## Top Level – All Systems

- For each system, show either
  - Average data for the interval
  - Average data (as a single point) for all intervals

## System Level

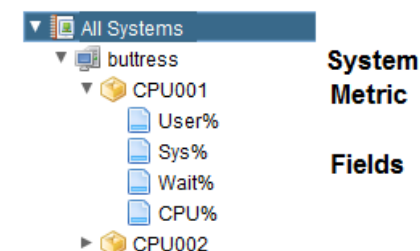
- For each system show data across time

## Metric Level

- For each metric show data across time for all fields

## Field Level

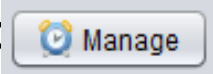
- For each field show data across time



**All reports & charts can be filtered by time using Intervals**

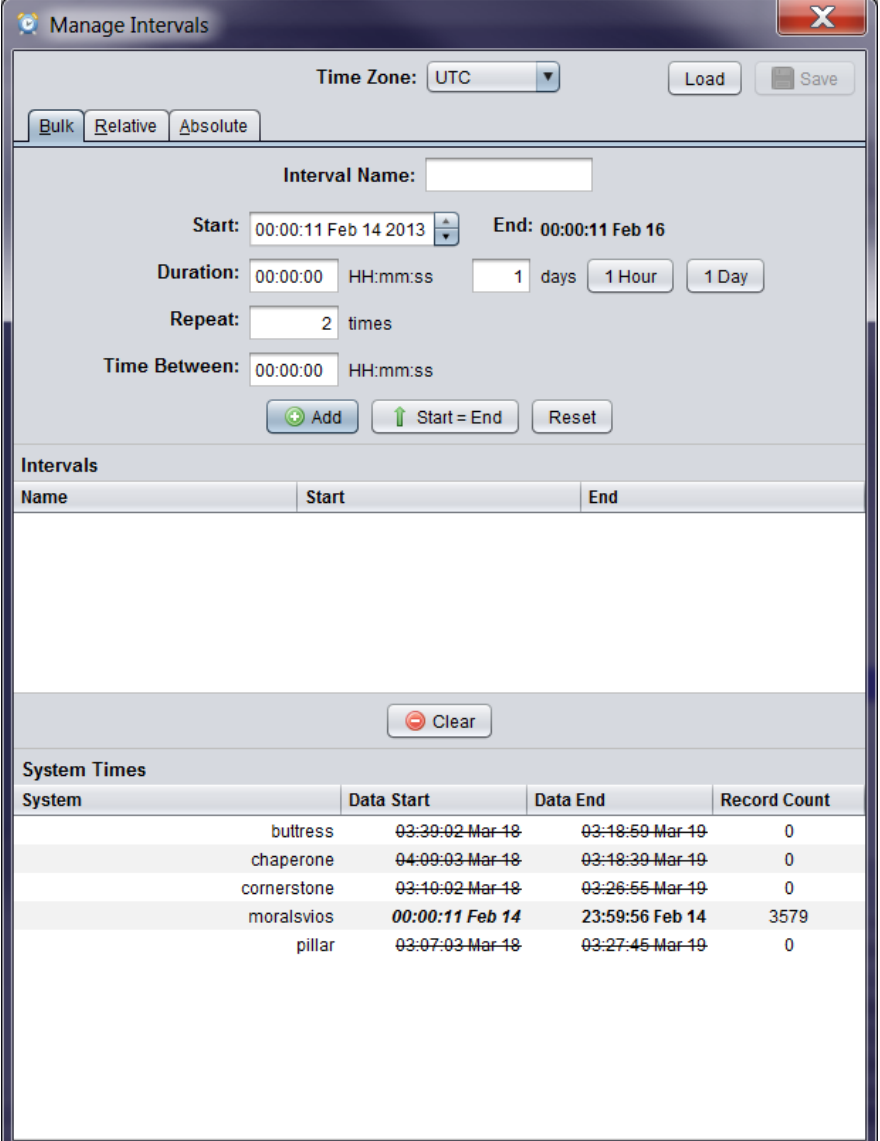
# Creating Intervals

## Open the Interval Manager dialog

- Click on Manage (the clock icon) 
- CTRL-I

## Supports multiple methods

- Absolute
- Relative to the earliest time in all files
- Bulk
  - Add multiple intervals over a given range
  - Supports separation between intervals (i.e. rampup)
  - Easy to add intervals that cover an hour or day via the 1 Hour / 1 Day buttons



Name	Start	End

System	Data Start	Data End	Record Count
bultress	03:39:02 Mar 18	03:18:59 Mar 19	0
chaperone	04:09:03 Mar 18	03:18:39 Mar 19	0
cornerstone	03:10:02 Mar 18	03:26:55 Mar 19	0
moralsvios	00:00:11 Feb 14	23:59:56 Feb 14	3579
pillar	03:07:03 Mar 18	03:27:45 Mar 19	0

# Interval Manager Tips & Tricks

## Charts are 'live'

- When an interval is added, the charts will update
- In the Interval Manager, right clicking on the interval will allow you to make the selected one the current
- Useful if you move the Interval Manager so it is not over the main window

## System Times section of the Interval Manager graphically indicates which systems have data for the current interval

- ~~Strikethrough~~ – no data
- ***Bold Italics*** – interval exactly matches start and / or stop
- **Bold** – interval is before start or after stop
- This data updates as you change the start / stop in the interval dialog or click on an existing interval

## Right clicking on a system will allow either

- Setting the start and stop to the system time
- Creating a new interval directly

System Times			
System	Data Start	Data End	Record Count
buttress	<del>03:30:02 Mar 18</del>	<del>03:18:50 Mar 19</del>	0
chaperone	<b><i>04:00:03 Mar 18</i></b>	<b><i>03:18:30 Mar 19</i></b>	0
cornerstone	<b>03:10:02</b>	Use as Start & End Mar 19	0
moralsviov	<b>00:00:11</b>	Add Interval Feb 14	3579
pillar	03:07:03 Mar 18	03:27:45 Mar 19	0

## Changing the Current Interval

**Once intervals are created, you can switch between them with**

- The Intervals menus
- The Interval drop down at the top of the UI
- CTRL-ALT-1 through CTRL-ATL-9 switch between the first 9 intervals

**There is always a default, All Data interval that will display all the parsed data**

- Unique entry in menu and drop down
- CTRL-ALT-0

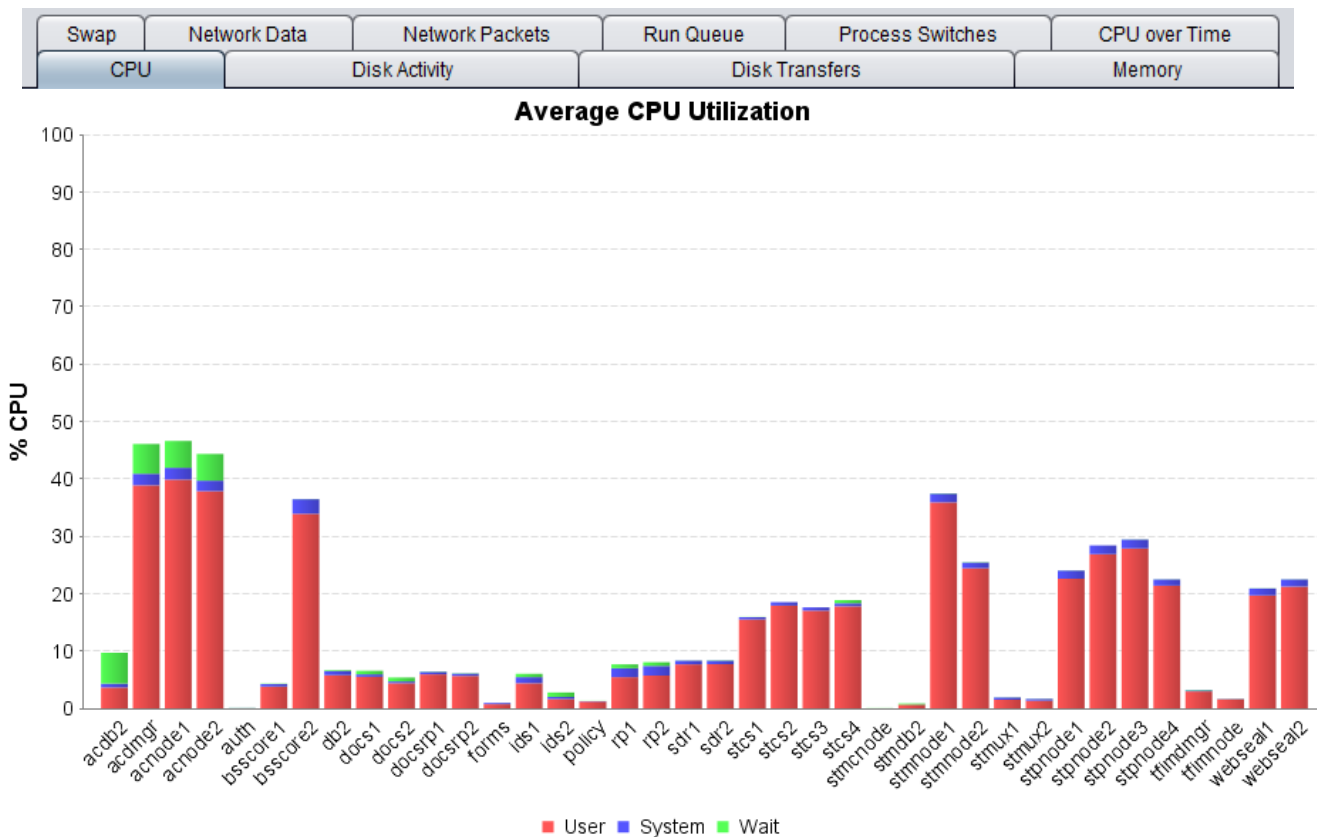
**Charts are 'live' and update as the interval changes**

- Application title also changes

# Where to start the analysis

## All Systems report shows average, summary data

- Does one system stand out with higher CPU, memory disk or network?
  - Is it what you expected?



## Digging into a specific system

Page Faults	Paging Amount	CPU by Process	All CPUs	CPU Balance	Disk Space	
Disk Busy	Disk Transfers	Network Reads	Network Writes	Run Queue	Process Switches	
CPU	Disk	Memory	Swap	Network	Disk Reads	Disk Writes

### Individual System reports show data over time

- CPU – spikes in the middle of the run / plateau?
- Memory & Swap – expected memory usage and no swap?
- Disk Reads / Writes – IOs on the expected disks?
- Disk Busy / Disk Transfers – IO rate is not too high?
- Network Reads / Writes – traffic on the expected interfaces?
- CPU by Process – expected processes are using CPU?
- CPU Balance – CPU utilization is balanced across cores?  
Hyperthreads / SMT threads use is lower at low overall CPU?

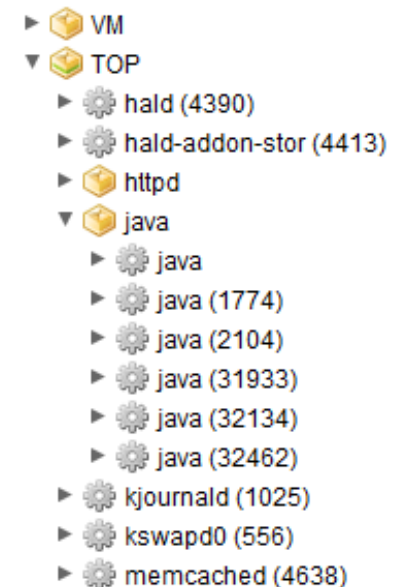
# Raw nmon Data

## All nmon data is available in the Metric & Field level charts

- Some disk data is duplicated (dm-x vs associated sdy's)
- Some disk data is additive ( $sdx = sdx1 + sdx2 + \dots + sdxn$ )

## The most useful data is probably the per process data, TOP

- Different organization than other data
  - Entry per process when there is a single instance of the process
  - Folder per process when there are multiple instances



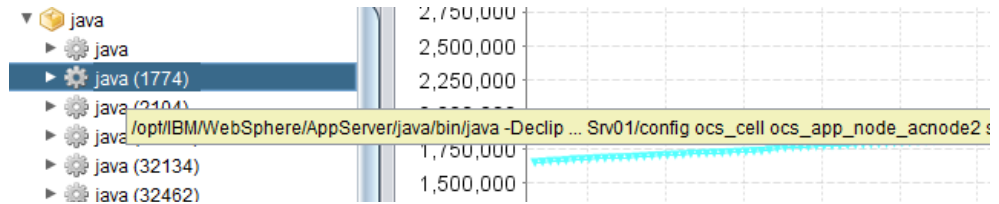
- Aggregated data is in the process without

a pid

# TOP Data Tips & Tricks

## Hover over a process in the tree

- A tool tip will display a shortened version of the command line



## Right click on a process

- Process Info option opens a dialog with the full command line
  - Each command line option is on a separate line
  - Data is selectable for copy & paste
- Note process start and stop time may not be accurate
  - Start time may be when nmon was started
  - Stop time may be when nmon was stopped
- Link with Tree option can be unchecked



- Open a second process info dialog to



# Chart Tips & Tricks

## **Click on a line / bar and it will highlight the row in the table**

- ... and vice-versa
- Multi-selection is supported
- Clicking on the legend item also works
  - This may be easier since lines actually have to be clicked on the data point

## **Mouse over is also supported**

- Line charts show series name, time and value
- Bar charts show series name, value and percentage of total

## Chart Tips & Tricks

### **Right click on the table or CTRL-SHIFT-C to change columns**

- CTRL-A for all; CTRL-D for defaults; CTRL-X for none
- This sets for all charts at the same 'level', e.g. only System charts

### **Line charts have an option to show / hide selected lines**

- Click the checkbox
- Right click on table
  - Show selected, show none or show all
- These are saved per-metric (i.e. CPU001 != CPU002)

### **Tables are sortable by column**

- This sets for all charts at the same 'level'

# Granularity

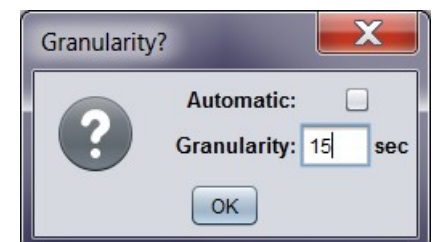
## All charts as displayed with a globally set 'granularity'

- How granular are the data points on the chart?
- Sets how much data is averaged to compute a single point on a line chart
- Measured in seconds
- Concept similar to Load Runner analysis

## Automatically calculated by default

- Fit ~100 points on a chart
- Updates as intervals change

Manually set via View → Set **Granularity** or CTRL-G



# Data Tables

## Data tables as a way to create summary, spread sheet like views of data

- Click on the Table radio button, View → Summary Table, or CTRL-T



## Drag metrics or fields from the tree onto the data table

- Metric will include all fields
- Automatically analyzes data for all systems

## Remove data with right click option or Clear button

- Remove option removes field for all hosts

# Data Tables, continued

## Two views supported

- Default shows a row per host / data type / metric combo with statistics for columns
- 
- Transposed shows a row per data type / metric combo with hosts for columns
  - Single displayed statistic is selectable via a drop down
- 
- Change via Transpose button or CTRL-SHIFT-T

Hostname	Data Type	Metric	Minimum	Average	Maximum	Std Dev	Sum
buttress	CPU_ALL	Busy	0.000	0.000	0.000	0.000	0.000
chaperone	CPU_ALL	Busy	0.000	0.000	0.000	0.000	0.000
cornerstone	CPU_ALL	Busy	0.000	0.000	0.000	0.000	0.000
moralsviov	CPU_ALL	Busy	0.000	0.000	0.000	0.000	0.000
pillar	CPU_ALL	Busy	0.000	0.000	0.000	0.000	0.000
buttress	CPU_ALL	CPU%	4.400	4.513	6.400	0.242	12,811.300
chaperone	CPU_ALL	CPU%	2.500	3.171	12.500	0.691	8,804.500
cornerstone	CPU_ALL	CPU%	4.700	8.682	15.300	2.026	25,272.400
moralsviov	CPU_ALL	CPU%	0.600	8.034	19.900	7.263	28,754.500
pillar	CPU_ALL	CPU%	0.500	0.577	2.500	0.070	1,685.200
buttress	CPU_ALL	CPUs	24.000	24.000	24.000	0.000	68,136.000
chaperone	CPU_ALL	CPUs	24.000	24.000	24.000	0.000	66,648.000
cornerstone	CPU_ALL	CPUs	32.000	32.000	32.000	0.000	93,152.000
moralsviov	CPU_ALL	CPUs	4.000	4.000	4.000	0.000	14,316.000
pillar	CPU_ALL	CPUs	32.000	32.000	32.000	0.000	93,408.000
buttress	DISKREAD	sda	0.000	0.202	11.500	0.253	573.300
chaperone	DISKREAD	sda	0.000	0.538	939.500	18.197	1,493.600
cornerstone	DISKREAD	sda	0.000	0.000	0.000	0.000	0.000
pillar	DISKREAD	sda	0.000	0.000	0.000	0.000	0.000
buttress	CPU_ALL	Sys%	4.300	4.356	5.400	0.062	12,366.500
chaperone	CPU_ALL	Sys%	1.600	2.141	4.400	0.251	5,944.400
cornerstone	CPU_ALL	Sys%	3.200	6.759	10.900	2.028	19,675.000
moralsviov	CPU_ALL	Sys%	0.400	7.823	18.800	7.258	27,998.300
pillar	CPU_ALL	Sys%	0.300	0.302	1.400	0.024	881.700
buttress	CPU_ALL	User%	0.100	0.157	1.900	0.204	444.800
chaperone	CPU_ALL	User%	0.800	1.030	8.900	0.587	2,860.100
cornerstone	CPU_ALL	User%	1.200	1.923	5.200	0.434	5,597.400
moralsviov	CPU_ALL	User%	0.200	0.211	9.800	0.171	756.200
pillar	CPU_ALL	User%	0.200	0.275	1.300	0.060	803.500

Transpose

Data Type	Metric	buttress	chaperone	cornerstone	moralsviov	pillar
CPU_ALL	User%	0.157	1.030	1.923	0.211	0.275
CPU_ALL	Sys%	4.356	2.141	6.759	7.823	0.302
CPU_ALL	Wait%	0.008	0.023	0.003	0.004	0.000
CPU_ALL	CPU%	4.513	3.171	8.682	8.034	0.577
CPU_ALL	Busy	0.000	0.000	0.000	0.000	0.000
CPU_ALL	CPUs	24.000	24.000	32.000	4.000	32.000
DISKREAD	sda	0.202	0.538	0.000	N/A	0.000

# Saving & Exporting Charts

## Save to PNG

- Right click on a chart & select Save Chart
  - Prompts for a file name, suggests a sensible default
- Right click on the tree
  - Asks for a folder and uses the default file name
  - All Systems and individual systems save *all* charts in the report
- Fixed resolution of ½ of 1920x1080

## Copy to clipboard

- Right click on a chart & select Copy

## Export to CSV

- Right click on a chart & select Copy Chart Data
- Paste into a file, Excel Etc

# Saving & Exporting Raw Data as CSV

## Save to file

- Right click on system, metric or field in the tree, select Save to CSV

## Copy to clipboard

- Right click on system, metric or field in the tree, select Copy

## Data tables can also be copied by right clicking

- Copy ⇒ currently selected row
- Copy All ⇒ all rows
- Copy button ⇒ all rows
-

# Running nmon as a Service



# Running nmon as a service in RHEL 6

**Copy nmon initd script to /etc/init.d/nmon**

**Copy nmon logrotate script to /etc/logrotate.d/nmon**

**Create symbolic link from nmon executable to /usr/bin/nmon**

**Setup permissions & start nmon service**

```
chown root:root /etc/init.d/nmon
chmod 755 /etc/init.d/nmon
chown root:root /etc/logrotate.d/nmon
chcon --reference /etc/init.d/network /etc/init.d/nmon
chmod 644 /etc/logrotate.d/nmon
chown root:root /usr/bin/nmon_x86_64_rhel6
chcon --reference /etc/logrotate.d/yum /etc/logrotate.d/nmon
chmod 755 /usr/bin/nmon_x86_64_rhel6
ln -s -f /usr/bin/nmon_x86_64_rhel6 /usr/bin/nmon
chkconfig --add nmon
service nmon start
```

**nmon collects every 30 seconds; 15 days of logs are retained in /var/log/nmon/old**

# Linux nmon Scripts

**See <http://nmonvisualizer.github.io/nmonvisualizer/startup.html>**

# Running nmon as a service in AIX

**AIX supports daily nmon collection out of the box.**

**Run smit topas to start smit.**

- **Select** Start New Recording → Start Persistent local recording
- **Select** nmon as type
- **Set the following options**
  - Recording interval in seconds: 30
  - Number of Days to retain: 15
  - Include Fiber Channel Section: yes (on VIOS)
  - Include Shared Ethernet Section: yes (on VIOS)
  - Include Large Page Section: yes
  - Include Asynchronous IO: yes
- Hit enter to run the command
- Esc 0 to exit smit

**There should now be an nmon file in /etc/perf/daily.**

**To stop nmon section, run smit topas. Then select** Stop Recording → Stop Persistent Recording  
→ Stop Persistent local nmon Record