Transforming Time Series Data into Capacity Planning Information

James F Brady
Capacity Planner for the State of Nevada
jfbrady@admin.nv.gov

Often an analyst has time series data available from performance monitors and needs to make statistical sense of it for capacity planning purposes. For example, a twenty-four hour column chart produced by averaging multiple days of time interval samples yields a statistically stable view of a resource's usage characteristics across the day and clearly identifies its busy period. Since monitoring tools often provide little support for this type of analysis, what can analysts do on their own to accomplish the needed data transformation? This paper describes valuable statistical manipulations and suggests approaches for capacity planning using "home grown" methods.

1.0 Introduction

Resource consumption data in time series format [WIKI13] is vital information when troubleshooting a performance problem with a system or network connection but has limited applicability in its native form for capacity planning purposes. Planning for the future requires aggregation of the data into a statistical structure where the underlying performance characteristics of a resource are unveiled and its busy period is clearly delineated. This aggregation requires the timestamp data be transformed into a time of day structure yielding graphical and tabular results that are statistically stable and useful for long range planning.

Figure 1 illustrates this transformation for the percent of resource utilization using data from the first twenty-eight days in April, 2013. The graph on the left is a time series plot of hourly observations for that interval while the graph on the right is a twenty-four hour distribution chart of the data with weekends removed and all remaining observations averaged by hour of the day. The right hand graph is a statistically stable view of resource utilization over the month and clearly identifies the busy period of the day, making it useful capacity planning information.

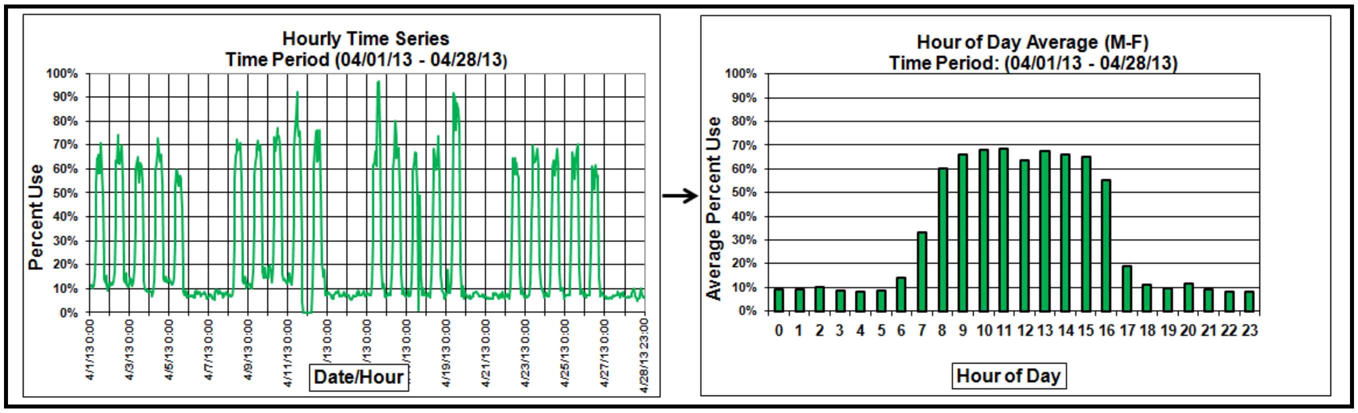


Figure 1: Time Series Data Transformed Into an Hour of Day Distribution

Monitoring tools often do not contain the functionality to produce the right hand chart in Figure 1 and there are at least three reasons why this is the case:

1. Most performance monitoring practitioners are perceived to be short term problem solvers not interested in long term capacity planning issues.
2. Users are thought to be uncomfortable selecting the number of data days to use when constructing the twenty-

four hour distribution chart because robust results depend on making a wise choice. A sample that is too small yields erratic results and one too large misses fundamental shifts in resource congestion behavior.

3. There is a concern that users unfamiliar with statistical inference and its nuances will misinterpret the information produced and reject the entire product for generating unreliable statistical information.

Given the lack of monitoring tool support, what can analysts do on their own to produce the right hand side of Figure 1? The following is intended to address this question using home grown methods to transpose the monitor data into this type of statistically useful graph.

The paper begins by defining the scope of the capacity planning information produced in this home grown environment. This is followed by a specific example of the Figure 1 data transformation process including sample size selection, capacity level determination, and trending over multiple sampling intervals. This illustration leads to a second example where column chart construction is performed in a more complex data collection and analysis environment. Next is a sample size selection discussion from both historical and experiential perspectives, and finally, some conclusions are drawn based on the ideas and illustrations presented.

2.0 Capacity Planning Scope

There are many aspects to the capacity planning process but within this discussion the focus is on creating the information required to establish an individual resource's current utilization level, its capacity limit, and when that limit is reached. The analysis needed to produce these results consists of four steps:

1. Transform the time series data into time of day statistics as in Figure 1.
2. Determine the busy period during the day by inspecting the column chart produced.
3. Use busy period load, sample size, and time of day to specify capacity limits.
4. Trend busy period samples to project when capacity will be reached.

The software used to process the data in the examples below is a set of Perl scripts [Perl13] developed by this author which produce column charts in portable network graphics, png, format and generate comma separated value, csv, files imported into pre-structured spreadsheets. Perl is chosen for this task because of its data parsing capabilities, its inclusion as part of Unix/Linux environments, and its availability within a free community edition. The scripts and examples discussed in this paper are also available free as a "grow your own" instructional starter kit. Email this author if you are interested.

3.0 Data Network Circuit Example

The four capacity planning steps are first illustrated with a data network circuit where a month of hourly percent utilization data is analyzed. Data transformation for this network element begins with Figure 2 which contains a time series graph of that circuit's incoming and outgoing percent utilization for the first twenty-eight days of April, 2013.

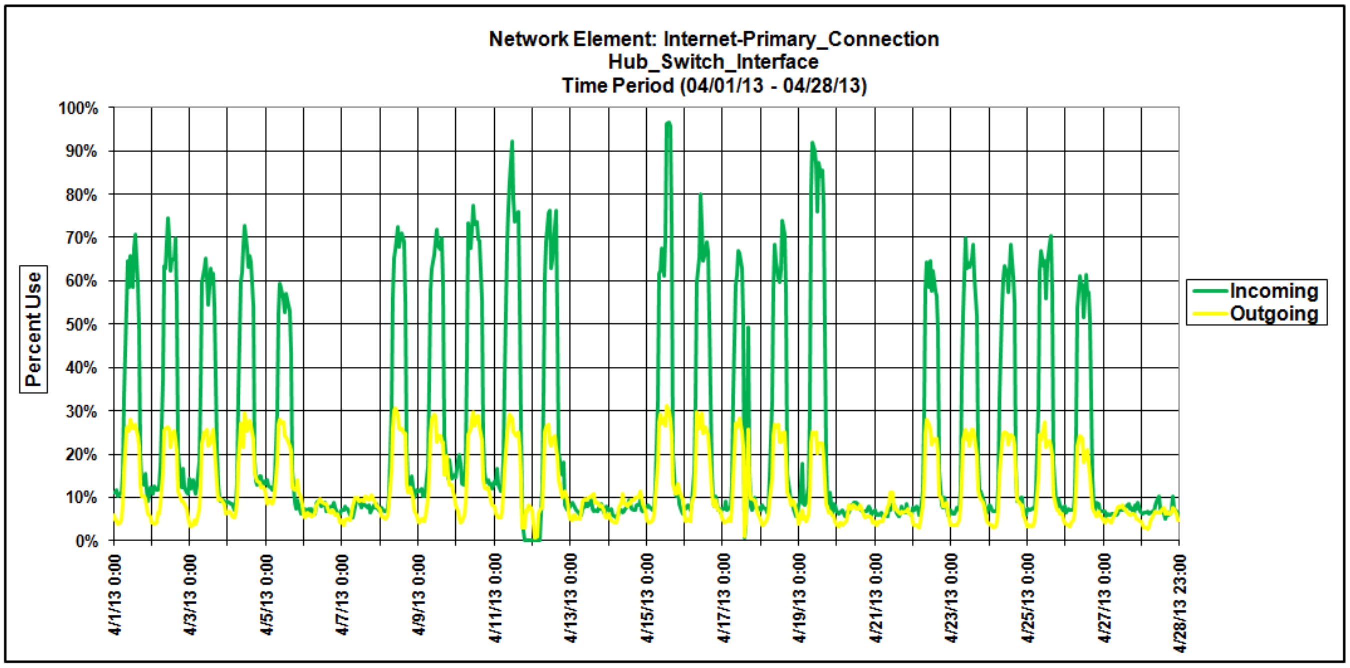


Figure 2: Percent BW Used Time Series Data for Data Network Circuit in April 2013

Inspection of this figure indicates the incoming side of the circuit is the most congested with spikes ranging from 70% to over 90% busy. It is difficult from this time series representation of the data to identify what hour of the day is consistently the busiest and the most important from a capacity planning perspective.

Figure 3 provides the needed statistical clarity with png graphs containing the average and peak hour percent bandwidth used excluding weekends. This set of charts illustrate this circuit is prime time oriented (8:00 AM – 5:00 PM) with its incoming traffic much greater than outgoing. These traffic directionality and timing characteristics are consistent with this network element's role as an internet circuit carrying local traffic.

The Figure 1 illustration is a combination of the incoming side of Figure 2 and the upper left hand corner graph in Figure 3. The right side of Figure 3 provides a rudimentary indication of data dispersion from the average by showing the peak hour of day value over the month for incoming (top) and outgoing (bottom) traffic.

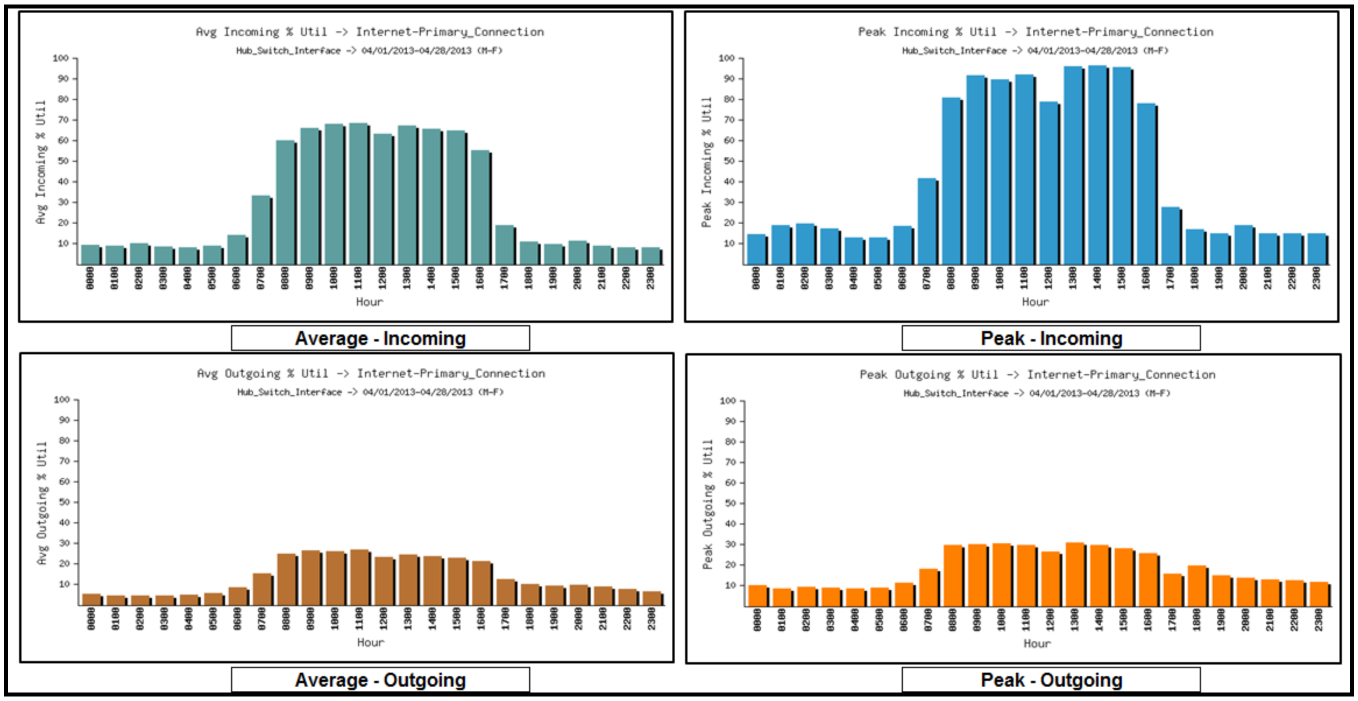


Figure 3: Average and Peak Hourly Percent BW Used png Graphs For Inc (Top) and Out (Bottom) Traffic

Figure 4 is the Figure 3 information imported into a pre-structured spreadsheet from a csv file where incoming and outgoing percent use are displayed together while average and peak are separated. The left side of this chart indicates the average busy hour occurs from 11:00 AM to 12:00 PM for the incoming (dominant) direction and the utilization at that time is 69%. This busy hour falls in the prime time window where traffic is on-line user based and cannot be scheduled. The right side of this figure provides the same rudimentary indication of data dispersion from the average as that shown in the right side of Figure 3 with incoming and outgoing combined on the same chart.

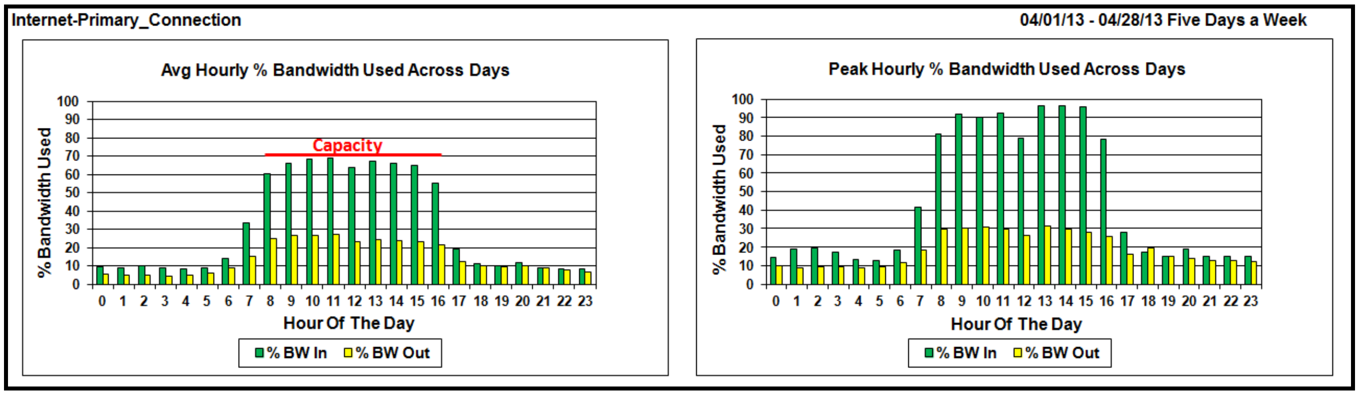


Figure 4: Average and Peak Hourly Percent BW Used Spreadsheet Graphs with Inc and Out Traffic Combined

The left side of Figure 4 also displays a "Capacity" line set to 70% which spans the 8:00 AM to 5:00 PM interval. Since the busy hour nearly touches that line the circuit is very close to capacity for the month. Viewed together, the two graphs in Figure 4 show that when the busy hour average of the twenty business day sample is around 70% the peak hour usage is close to 100%. For the circuits contained in the environment from which the network element sample is drawn this "70% rule of thumb" works pretty well but is obviously a judgment call.

The results in Figure 4 represent a sample of size one and provide no real history of the prime time busy hour average usage level and where it is trending into the future. Figure 5 provides this insight by charting the last twelve months of these values, the trend line associated with them, and the 70% capacity line. This set of graphs indicate the prime time busy hour average in both March and April of 2013 are significant shifts upward from the previous ten months so some investigation is warranted before concluding that an immediate upgrade is justified.

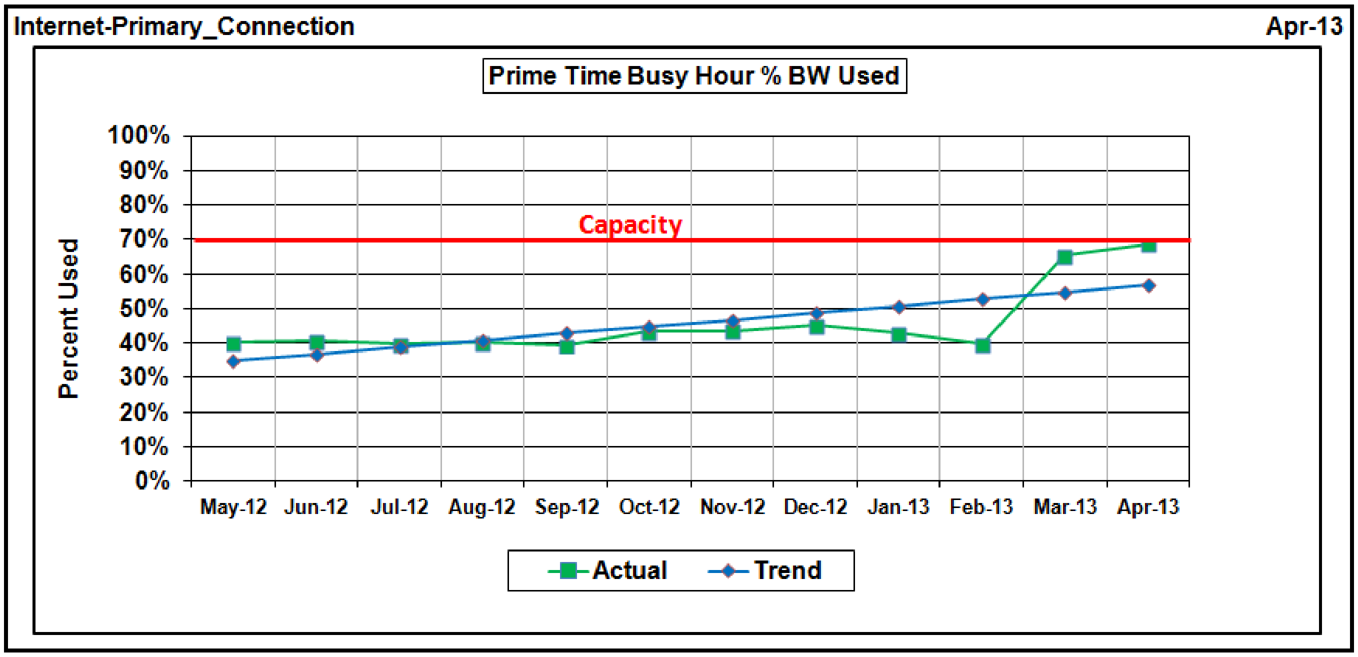


Figure 5: Prime Time Busy Hour Percent BW Used Over the Last Year.

All four capacity planning goals mentioned in Section 2.0 have been accomplished. The time series data has been translated into time of day statistics, the busy period during the day identified, a capacity limit determined, and a trend line produced to estimate the capacity exhaustion point.

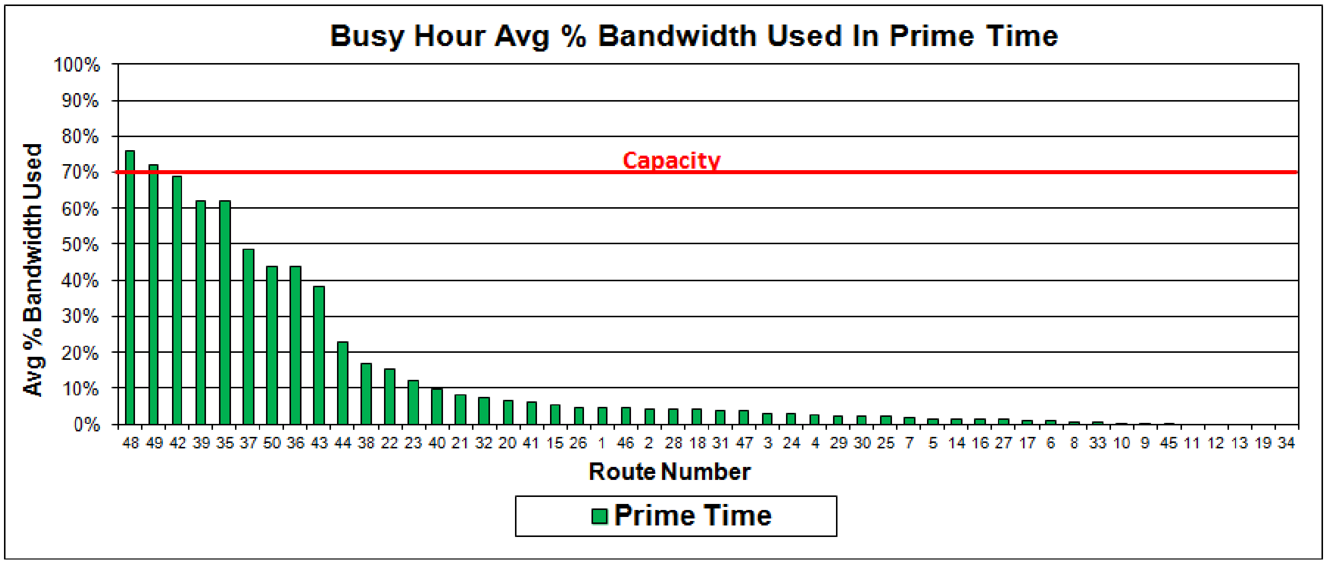
Appendix A contains an overview of the Perl script used to process the input file for this example including the Figure A1 flow diagram showing that file's layout and an illustration of the png and csv output produced. From an operational perspective the png files, Figure 3, are imported into various capacity planning and upgrade justification documents as part of the overall network planning process. The csv data is imported into a spreadsheet where fifty circuits are analyzed in this manner yielding a set of Figure 4 graphs for each.

The fifty circuit spreadsheet also contains the Figure 6 summary worksheet sorted by prime time busy hour average percent use in the dominant direction, i.e., Prime Time Max % Use. This figure additionally shows the 70% capacity line and the first three rows of a table containing circuit usage details. These details are the worksheet Tab number used to locate the circuit's Figure 4 graphs, Route ID, Bandwidth in Mbits/sec, Busy Hour Inc % Use, Busy Hour Out % Use, Prime Time Max % Use, Prime Time Use Rank, and Use Type. Busy Hour Inc % Use and Busy Hour Out % Use are based on a twenty-four hour period with Use Type specified as Prime Time when one of their values is equal to the Prime Time Max % Use.

The prime time busy hour is the most critical time period from a circuit ranking perspective because that traffic is generally user demand based transactions which cannot be scheduled. Most workloads that occur during non-prime time hours are composed of schedulable traffic such as backups.

The network elements in this list of fifty are trended and ranked in a top ten list spreadsheet with graphs like Figure 5 produced in separate worksheets for each. These trend charts represent the circuit's prime time busy hour utilization level over the long term and provide an indication when capacity will be reached.

Network Elements **04/01/13 - 04/28/13 Five Days a Week**



			Busy Hour Use		Prime Time Use		
Tab	Route ID	Mbits/Sec	Inc %	Out %	Max %	Rank	Use Type
48	Network-Circuit_A	100	76.06%	0.76%	76.06%	1	Prime Time
49	Network-Circuit_B	100	71.94%	21.57%	71.94%	2	Prime Time
42	Internet-Primary_Connection	100	68.71%	26.97%	68.71%	3	Prime Time

Figure 6: Summary Worksheet for Fifty Circuits Sorted by Highest Prime Time Busy Hour Average Value

4.0 Virtual Server Guest Physical CPUs Used Example

The second example being discussed is an analysis of physical CPU's used by a single Unix guest in a virtualized server environment. The guest is running AIX Unix in an IBM pSeries server with automated data collection performed by Nmon [Nmon06], a free tool used to analyze AIX Unix and Linux performance on IBM pSeries and PureFlex systems. This tool collects a vast amount of resource consumption information at the hypervisor and individual guest level but a single guest's physical CPUs used is the only counter being analyzed.

This illustration provides some twists to the four steps discussed in the previous exercise regarding data transformation, the capacity statement, and the trending of busy period observations. For example, the data collection portion of data transformation is performed every fifteen minutes and stored in daily files. This is quite different from the previous illustration where data is reported hourly and included in a single monthly file for a set of network circuits.

Figure 7 shows two days of Nmon time series graphs for the Sysxxp00 virtual guest where the number of physical processors used is plotted in fifteen minute intervals for a twenty-four hour period. These graphs indicate the largest number of CPUs is consumed during prime time with another utilization surge from 7:00 PM to 10:00 PM. This is a typical traffic pattern for this type of server which handles user transactions during the day and performs backups in the early evening.

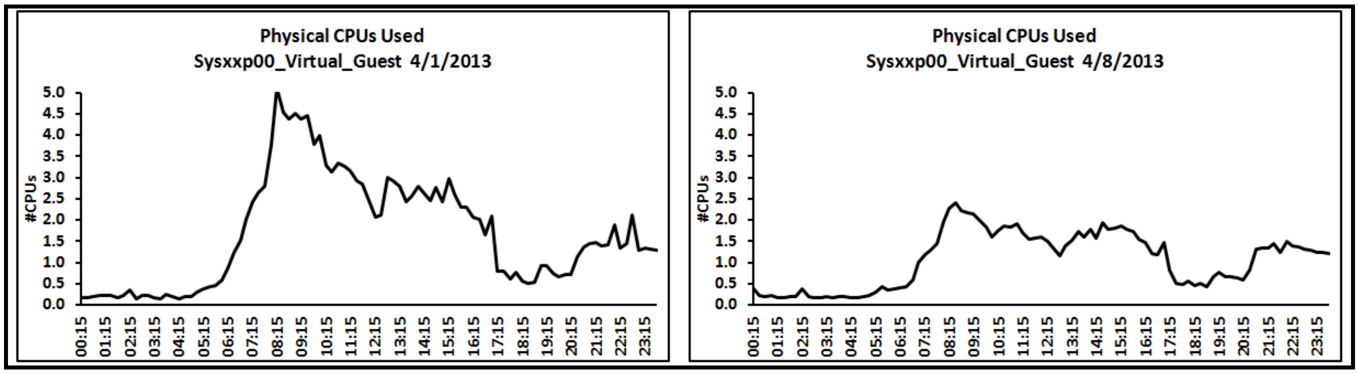


Figure 7: Virtual Guest Physical CPUs Used Time Series Data for Two Mondays

These charts provide useful information regarding CPUs used for the specific days represented, but being a sample of size two, they don't yield sufficient insight into CPU consumption for capacity planning purposes. The contrast in number of CPUs used on the right vs left side of the figure supports this insufficiency argument, especially since both graphs represent the same day of week while displaying such load diversity.

Figure 8 is a capacity planning oriented graph showing the average and peak hour number of CPUs used in png graph format for the first twenty weekdays in April 2013. This graphical set confirms the earlier assertion that this processing environment is prime time oriented with another small surge in load during the early evening. As is the case with Figure 3, the right side of Figure 8 provides a rudimentary indication of data dispersion from the average by showing the peak hour of day over the month. The vertical axes range of values is different between the two figures since Figure 8 provides a count and Figure 3 lists a percentage.

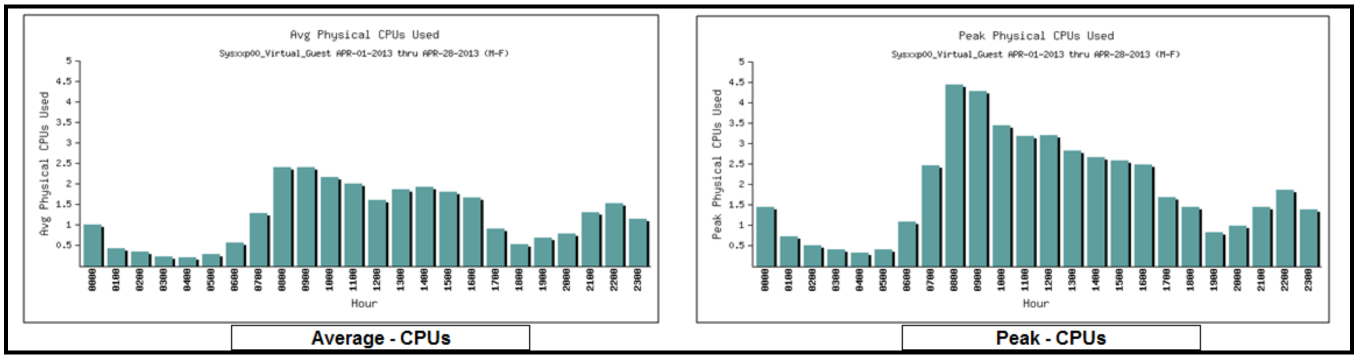


Figure 8: Virtual Guest Avg and Peak CPUs Used for First Twenty Weekdays in April 2013

Figure 9 is the Figure 8 information imported into a pre-structured spreadsheet from a csv file produced by a Perl script similar to the way Figure 4 is related to Figure 3. The left side of this chart indicates the average busy hour occurs from 8:00 AM to 9:00 AM and shows 2.40 physical CPUs being consumed at that time. This busy hour falls in the prime time window so it occurs during the critical busy period for this processing environment. The right side of this figure provides the same rudimentary indication of data dispersion from the average that Figure 4 does for the network circuit.

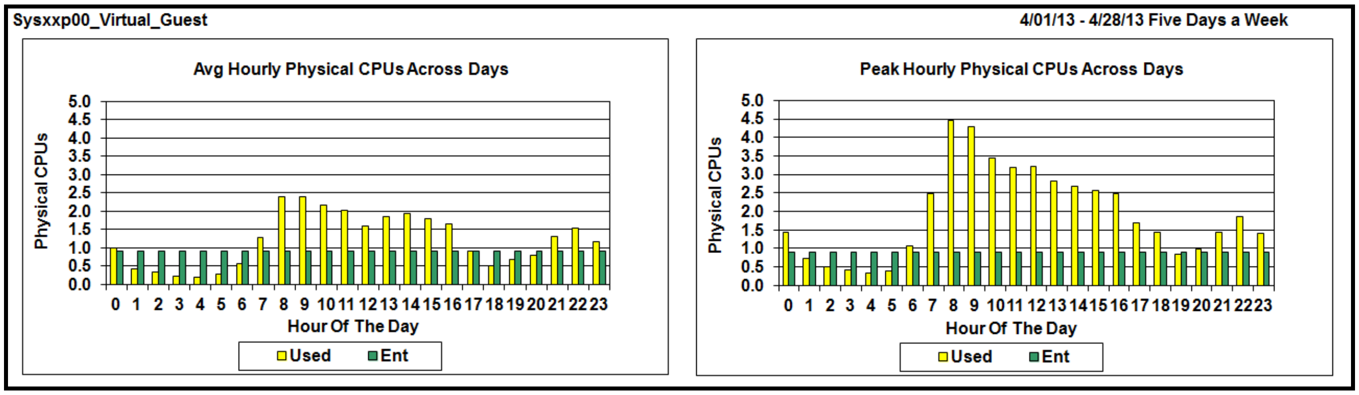


Figure 9: Avg and Peak CPUs Used and Entitled Spreadsheet Graphs for First Twenty Weekdays in April 2013

In this virtual world, capacity is more appropriately specified as a minimum resource allocation level than a fixed value so, rather than the capacity line shown in the left side of Figure 4, the Figure 9 charts display an entitlement of .90 physical CPUs. In AIX Unix, entitlement is an administratively set value representing the minimum number of CPUs allocated to the guest by the hypervisor from its pool of physical processors. This virtual guest exceeds its entitlement on a regular basis and can do so as long as CPU resources are available from the hardware pool or higher priority guests not currently using their entitlement. Because it is a critical transaction based processing environment, the analysis suggests the entitlement should be raised to the prime time busy hour average level of 2.4 physical CPUs.

The first three capacity planning steps, data translation, busy period identification, and capacity (entitlement) determination, have been accomplished for this example. The fourth step, trending busy period observations, is not incorporated into the current analysis structure but an effort is underway to do so using available Nmon data.

Appendix B contains an overview of the Perl script used to process the input files for this example including a flow diagram showing the daily input file list and a layout of the key Nmon records associated with physical CPUs used. The fifteen minute samples are averaged to produce hourly statistics and weekend files are optionally ignored if included in the input directory. Figure B1 illustrates the creation of png and csv files which are imported into various capacity planning and upgrade justification documents within the organization as part of the overall capacity planning and budgeting process.

The Perl script produces png and csv files for other resources monitored by Nmon including the CPU entitlement in Figure 9, real memory, disk I/Os, network packets, context switches, run queue size, and CPU pool size. There is no discussion of these output files here but examples of them are included as part of the free script package mentioned in Appendix B.

5.0 Sample Size Selection

Both examples use a twenty business day sample to produce capacity planning information but is this choice unique to these resource monitoring situations or is there a more fundamental basis for the sample size chosen? Certainly judgment plays a role but, unless there is some fundamental property of the traffic flow that can be exploited, a review of other sampling environments is instructive. The telephone network is such an environment where switching and connection component time series data is converted to hour of day statistics for busy period identification purposes. This author is familiar with this process as author of the GTE Traffic Grade of Service Standards [GTE85] referenced in that company's (now Verizon) Federal Access Tariff as its traffic sizing rule book in response to AT&T divestiture.

The traffic engineering rule for sizing speech paths (trunk groups) between switching systems, as an example, is based on the Average Busy Season Busy Hour (ABSBH) which is the highest traffic volume hour of the day determined by averaging hour of day traffic on a five day a week basis over the four highest contiguous weeks during the engineering period [Hay82] [Hil76]. This sampling rule, used to size billions of dollars worth of circuit switched resources, is the same concept as used to produce the busy hour average results in these examples. This sampling plan generally works well for this author because it tends to capture fundamental shifts in resource consumption levels across the day while damping out the noise associated with random fluctuations in the data. Special insight may dictate a different sampling rule but this is a good starting point.

6.0 Conclusions

Capacity Planners often complain there is no commitment from management to support their efforts while they are passively sitting on a gold mine of data which only needs to be transposed into the needed planning level information. This author frequently performs this type of data transformation for the internal customers within the organization and begins the effort by asking those monitoring the resources for readily available time series data without a commitment from them to spend time or money. Typically, the question is simply; "may I have a sample of the data being produced by the monitoring tool?" Once the Perl script is executing correctly with the sample, a full set of data is processed and results produced. The provider of the data, the management responsible for supporting the resource being analyzed, and the financial organization controlling the budget, see the benefits of the analysis and momentum is gained for the capacity planning process. Not only is the time series data converted to capacity planning information but the organization becomes more capacity planning oriented with the capacity analyst driving the process.

References

[Bra10] J.F. Brady, "Making Statistical Sense out of Time Series Data with 'Home Grown' Perl Scripts", CMG MeasureIT, (July 2010). http://www.cmg.org/measureit/issues/mit71/m_71_4.pdf

[GTE85] GTE Service Corporation Telephone Operations, "Traffic Grade of Service Standards", April, 1985.

[Hay82] W.S. Hayward and P.J. Moreland, "Theoretical and Engineering Foundations", The Bell System Technical Journal, Volume 62, Number 7, Sept, 1983.

[Hil76] D.W. Hill and S.R. Neal, "Traffic Capacity of a Probability-Engineered Trunk Group", The Bell System Technical Journal, Volume 55, Number 7, Sept, 1976.

[Nmon06] N. Griffiths, "nmon performance", (2006). http://www.ibm.com/developerworks/aix/library/au-analyze_aix

[Perl13] L. Wall, T. Christiansen, "Programming Perl", http://www.en.wikipedia.org/wiki/Programming_Perl

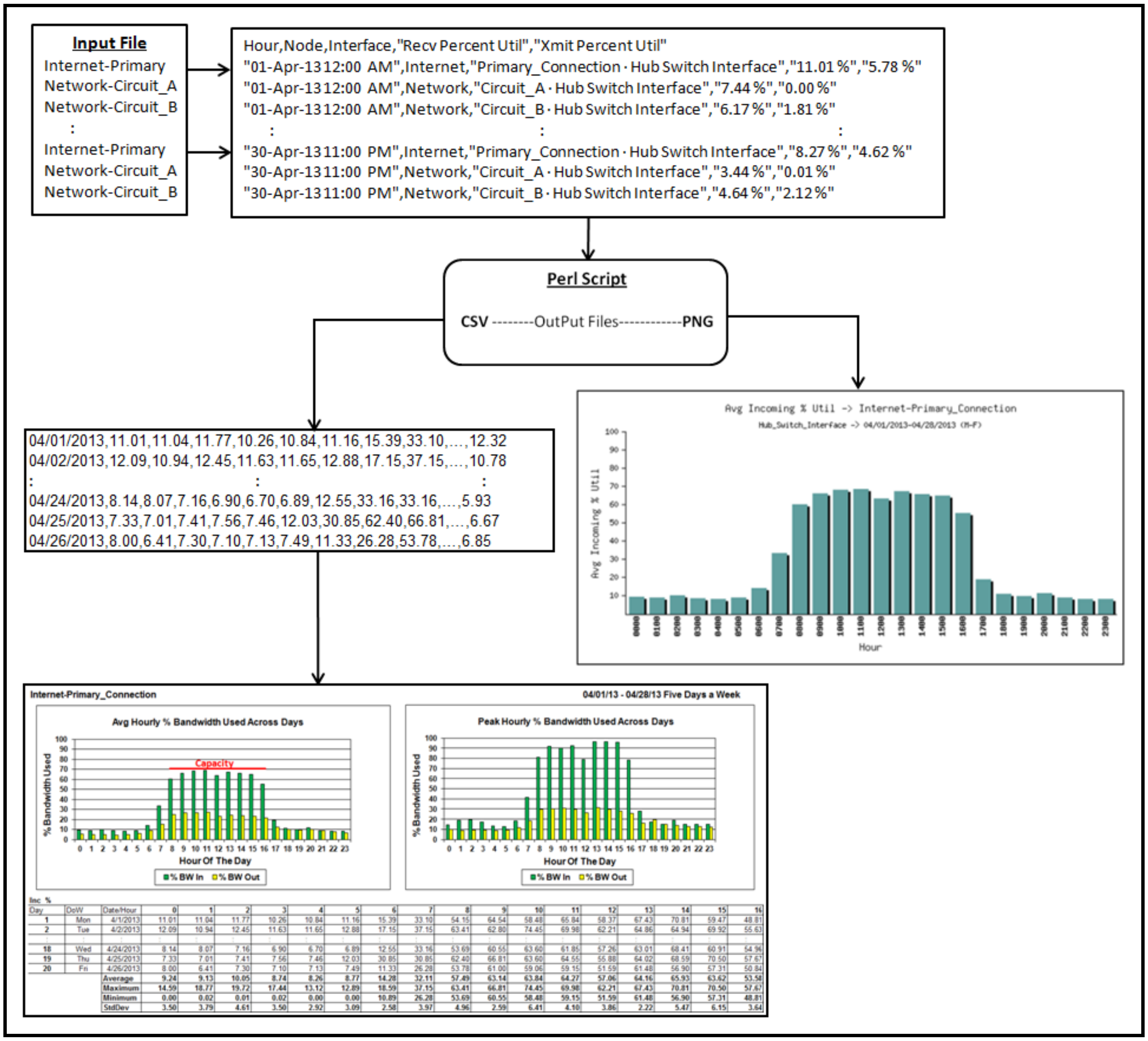
[WIKI13] Wikipedia, "Time series", (2013) http://www.en.wikipedia.org/wiki/Time_series

Copyrights and Trademarks

Appendix A
Data Network Circuit Perl Script

Figure A1 provides a functional flow diagram of the Perl script used to produce the csv and png results in Section 3.0. The script reads the network analysis tool's csv file containing a row of data for each circuit on an hour interval basis, extracts the data for weekdays, reorients that data to match the spreadsheet layout, and creates csv files like shown in the middle left of the figure. The csv files are imported into a spreadsheet set up to calculate statistics and display the graphs shown here and in Figure 4. The Perl script also generates a set of graphical column charts in png format like the one in the middle right of the figure. A full set of these charts for the Internet-Primary circuit are pictured in Figure 3.

The best way to gain an understanding of this analysis environment is to request the free copy of the example by emailing this author. The information returned is a WinZip file containing the Perl script, input file, output files, sample spreadsheet, and set of instructions describing script installation and execution.


Input File
Internet-Primary
Network-Circuit_A
Network-Circuit_B
:
Internet-Primary
Network-Circuit_A
Network-Circuit_B
Perl Script
CSV --------OutPut Files------------PNG
Avg Incoming % Util -> Internet-Primary_Connection
Hub_Switch_Interface -> 04/01/2013-04/28/2013 (M-F)
Avg Incoming % Util
Hour
Internet-Primary_Connection
04/01/13 - 04/28/13 Five Days a Week
Avg Hourly % Bandwidth Used Across Days
Capacity
% Bandwidth Used
Hour Of The Day
■% BW In ■% BW Out
Peak Hourly % Bandwidth Used Across Days
% Bandwidth Used
Hour Of The Day
■% BW In ■% BW Out

```
Hour,Node,Interface,"Recv Percent Util","Xmit Percent Util"
"01-Apr-13 12:00 AM",Internet,"Primary_Connection · Hub Switch Interface","11.01 %","5.78 %"
"01-Apr-13 12:00 AM",Network,"Circuit_A · Hub Switch Interface","7.44 %","0.00 %"
"01-Apr-13 12:00 AM",Network,"Circuit_B · Hub Switch Interface","6.17 %","1.81 %"
      :                                    :                                       :
"30-Apr-13 11:00 PM",Internet,"Primary_Connection · Hub Switch Interface","8.27 %","4.62 %"
"30-Apr-13 11:00 PM",Network,"Circuit_A · Hub Switch Interface","3.44 %","0.01 %"
"30-Apr-13 11:00 PM",Network,"Circuit_B · Hub Switch Interface","4.64 %","2.12 %"
```

```
04/01/2013,11.01,11.04,11.77,10.26,10.84,11.16,15.39,33.10,...,12.32
04/02/2013,12.09,10.94,12.45,11.63,11.65,12.88,17.15,37.15,...,10.78
:                                  :                                  :
04/24/2013,8.14,8.07,7.16,6.90,6.70,6.89,12.55,33.16,33.16,...,5.93
04/25/2013,7.33,7.01,7.41,7.56,7.46,12.03,30.85,62.40,66.81,...,6.67
04/26/2013,8.00,6.41,7.30,7.10,7.13,7.49,11.33,26.28,53.78,...,6.85
```

Inc %

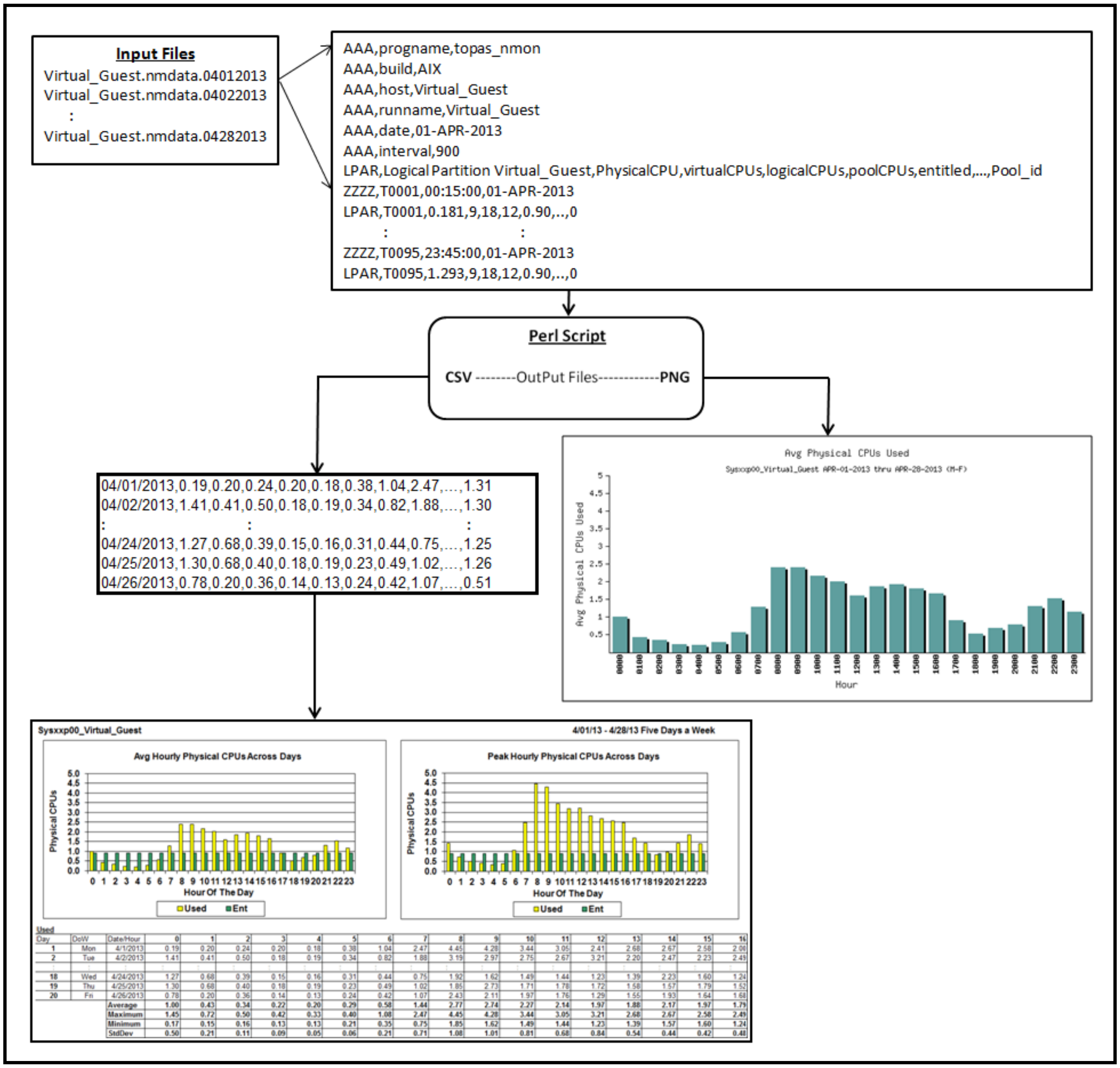
Day	DoW	Date/Hour	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	Mon	4/1/2013	11.01	11.04	11.77	10.26	10.84	11.16	15.39	33.10	54.15	64.54	58.48	65.84	58.37	67.43	70.81	59.47	48.81
2	Tue	4/2/2013	12.09	10.94	12.45	11.63	11.65	12.88	17.15	37.15	63.41	62.80	74.45	69.98	62.21	64.86	64.94	69.92	55.63
:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:
18	Wed	4/24/2013	8.14	8.07	7.16	6.90	6.70	6.89	12.55	33.16	53.69	60.55	63.60	61.85	57.26	63.01	68.41	60.91	54.96
19	Thu	4/25/2013	7.33	7.01	7.41	7.56	7.46	12.03	30.85	30.85	62.40	66.81	63.60	64.55	55.88	64.02	68.59	70.50	57.67
20	Fri	4/26/2013	8.00	6.41	7.30	7.10	7.13	7.49	11.33	26.28	53.78	61.00	59.06	59.15	51.59	61.48	56.90	57.31	50.84
		Average	9.24	9.13	10.05	8.74	8.26	8.77	14.28	32.11	57.49	63.14	63.84	64.27	57.06	64.16	65.93	63.62	53.58
		Maximum	14.59	18.77	19.72	17.44	13.12	12.89	18.59	37.15	63.41	66.81	74.45	69.98	62.21	67.43	70.81	70.50	57.67
		Minimum	0.00	0.02	0.01	0.02	0.00	0.00	10.89	26.28	53.69	60.55	58.48	59.15	51.59	61.48	56.90	57.31	48.81
		StdDev	3.50	3.79	4.61	3.50	2.92	3.09	2.58	3.97	4.96	2.59	6.41	4.10	3.86	2.22	5.47	6.15	3.64

Figure A1: Data Network Circuit Perl Script Functional Flow Diagram.

Appendix B
Virtual Guest Physical CPUs Used Perl Script

Figure B1 provides the functional flow diagram of the Perl script used to produce the csv and png results in Section 4.0. The script reads the daily Nmon files, extracts the data for weekdays, averages the fifteen minute samples over hour intervals, reorients that data to match the spreadsheet layout, and creates csv files like shown in the middle left of the figure. The csv files are imported into a spreadsheet set up to calculate statistics and display the graphs shown here and in Figure 9. The Perl script also produces a set of graphical column charts in png format like the one in the middle right of the figure and in Figure 8.

The best way to gain an understanding of this analysis environment is to request the free example by emailing this author. The information returned is a WinZip file containing the Perl script, input files, output files, sample spreadsheet, and a set of instructions describing script installation and execution. This executable demo also creates csv and png output for all of the other resources mentioned in the Section 4.0 discussion such as real memory.



Input Files
Perl Script
CSV --------OutPut Files------------PNG
Avg Physical CPUs Used
Sysxxp00_Virtual_Guest APR-01-2013 thru APR-28-2013 (M-F)
Avg Physical CPUs Used
Hour
Sysxxp00_Virtual_Guest
4/01/13 - 4/28/13 Five Days a Week
Avg Hourly Physical CPUs Across Days
Peak Hourly Physical CPUs Across Days
Physical CPUs
Hour Of The Day
Used
Ent

```
Virtual_Guest.nmdata.04012013
Virtual_Guest.nmdata.04022013
      :
Virtual_Guest.nmdata.04282013
```

```
AAA,progname,topas_nmon
AAA,build,AIX
AAA,host,Virtual_Guest
AAA,runname,Virtual_Guest
AAA,date,01-APR-2013
AAA,interval,900
LPAR,Logical Partition Virtual_Guest,PhysicalCPU,virtualCPUs,logicalCPUs,poolCPUs,entitled,...,Pool_id
ZZZZ,T0001,00:15:00,01-APR-2013
LPAR,T0001,0.181,9,18,12,0.90,..,0
         :                    :
ZZZZ,T0095,23:45:00,01-APR-2013
LPAR,T0095,1.293,9,18,12,0.90,..,0
```

```
04/01/2013,0.19,0.20,0.24,0.20,0.18,0.38,1.04,2.47,…,1.31
04/02/2013,1.41,0.41,0.50,0.18,0.19,0.34,0.82,1.88,…,1.30
:                   :                            :
04/24/2013,1.27,0.68,0.39,0.15,0.16,0.31,0.44,0.75,…,1.25
04/25/2013,1.30,0.68,0.40,0.18,0.19,0.23,0.49,1.02,…,1.26
04/26/2013,0.78,0.20,0.36,0.14,0.13,0.24,0.42,1.07,…,0.51
```

Used

Day	DoW	Date/Hour	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	Mon	4/1/2013	0.19	0.20	0.24	0.20	0.18	0.38	1.04	2.47	4.45	4.28	3.44	3.05	2.41	2.68	2.67	2.58	2.06
2	Tue	4/2/2013	1.41	0.41	0.50	0.18	0.19	0.34	0.82	1.88	3.19	2.97	2.75	2.67	3.21	2.20	2.47	2.23	2.49
:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:
18	Wed	4/24/2013	1.27	0.68	0.39	0.15	0.16	0.31	0.44	0.75	1.92	1.62	1.49	1.44	1.23	1.39	2.23	1.60	1.24
19	Thu	4/25/2013	1.30	0.68	0.40	0.18	0.19	0.23	0.49	1.02	1.85	2.73	1.71	1.78	1.72	1.58	1.57	1.79	1.52
20	Fri	4/26/2013	0.78	0.20	0.36	0.14	0.13	0.24	0.42	1.07	2.43	2.11	1.97	1.76	1.29	1.55	1.93	1.64	1.68
		Average	**1.00**	**0.43**	**0.34**	**0.22**	**0.20**	**0.29**	**0.58**	**1.44**	**2.77**	**2.74**	**2.27**	**2.14**	**1.97**	**1.88**	**2.17**	**1.97**	**1.79**
		Maximum	**1.45**	**0.72**	**0.50**	**0.42**	**0.33**	**0.40**	**1.08**	**2.47**	**4.45**	**4.28**	**3.44**	**3.05**	**3.21**	**2.68**	**2.67**	**2.58**	**2.49**
		Minimum	**0.17**	**0.15**	**0.16**	**0.13**	**0.13**	**0.21**	**0.35**	**0.75**	**1.85**	**1.62**	**1.49**	**1.44**	**1.23**	**1.39**	**1.57**	**1.60**	**1.24**
		StdDev	**0.50**	**0.21**	**0.11**	**0.09**	**0.05**	**0.06**	**0.21**	**0.71**	**1.08**	**1.01**	**0.81**	**0.68**	**0.84**	**0.54**	**0.44**	**0.42**	**0.48**

Figure B1: Physical CPUs Used Perl Script Functional Flow Diagram