Automating Performance Data Collection

Tim Clemens

netsysco
INFRASTRUCTURE SERVICES, LLC

Training and Consulting

Louisville, KY

1-800-888-9028

Topics

- Review of MeasureWare and Tools
- Express Goals for Enterprise Performance Data Collection
- Extracting Weekly Performance Data
- Exporting Weekly Summary Metrics
- Charting Summary Daily Automatically

Openview Performance Agent
MeasureWare Agent - MWA

- Collects running performance metrics
- Allows summary reports to be extracted
- Can be customized per application
- Can be configured to send alarms

Performance data collection

- scopeux daemon logs performance metrics to /var/opt/perf/datafiles
- Performance log files:

– logglob	Global metrics
– logappl	Application specific metrics
– logproc	“interesting” processes metrics
– logdev	Device metrics
– logtran	Transaction metrics
– logindx	Index for data in these files

- Data is collected every 5 minutes (process data is every minute)

MeasureWare Configuration

- Parameter file: /var/opt/perf/parm
- Control what is logged:

```
log global application process dev=disk,lvm transaction
```

- Control log file sizes:

```
size global=10, application=10, process=20,
devices=10, transaction=10
maintime=23:30
```

- “interesting” processes:

```
threshold cpu=5.0, disk=5.0, nonew, nokilled
```

- Define Applications:

```
application=oracle     or
file=ora*              group=dba
user=oracle
```

Performance Data Logs

- Amount of data retained is limited by size
- 25% of oldest data removed at `mainttime`

  (once size limit is exceeded)
- Global records are smaller - retains longer
- Process records are numerous

  Logged every minute instead of 5 minutes as with all other types
- Retained history data can vary greatly depending on system configuration and activity.

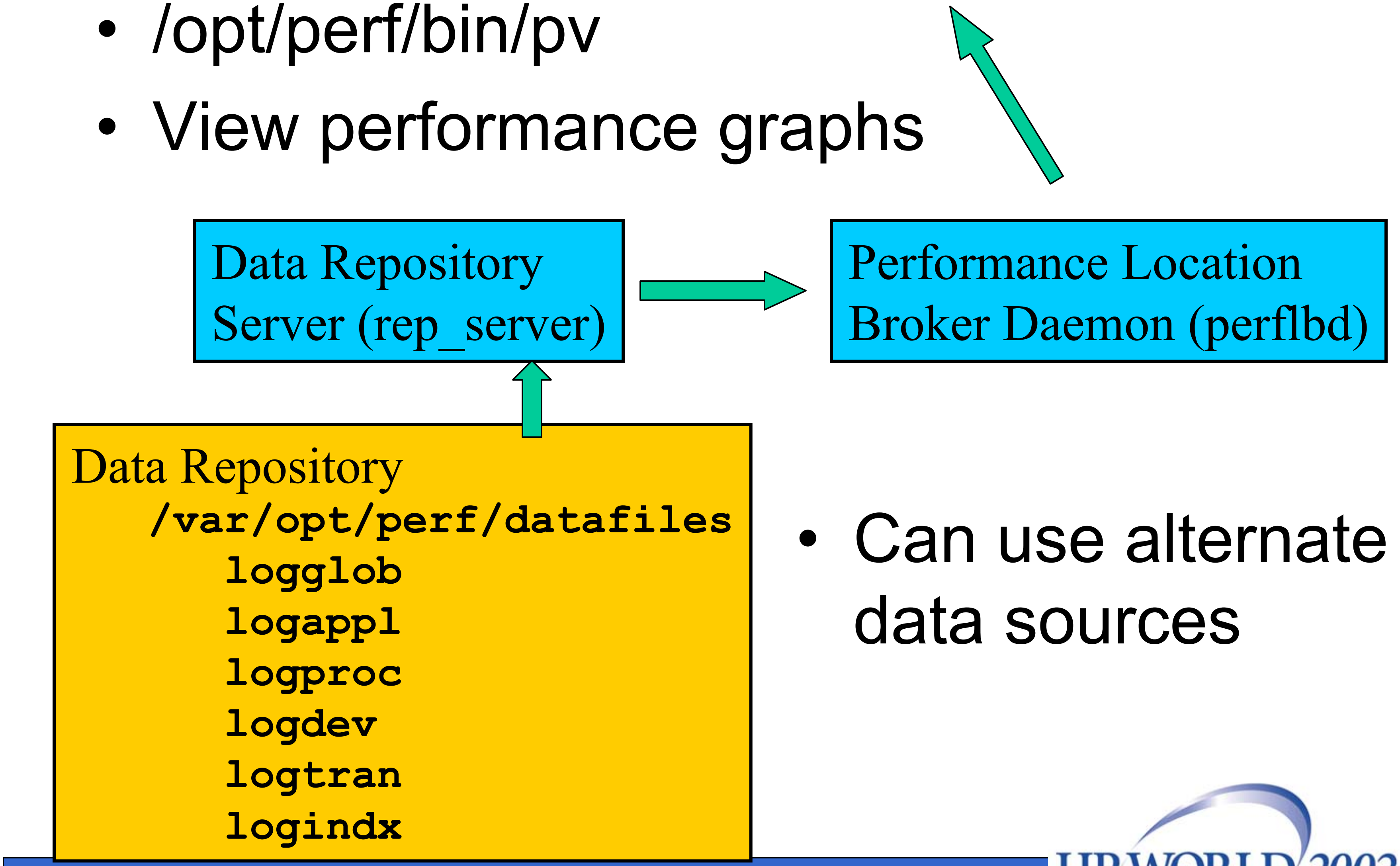
Determine Data Retention

- /opt/perf/bin/utility -xs

```
The total time covered was      : 164/11:59:35 out of 164/22:00:00
Time lost when collector was off:     10:00:25   0.25%
The scope collector was started :           31 times

            ------Total------  --Each Full Day-- -------Dates-------   Full
Type        Records MegaBytes  Records MegaBytes  Start       Finish   Days
Global        47409      9.54    288.2     0.055 01/30/02 to 07/13/02  164.5
Application   76669      9.52   3566.0     0.443 06/22/02 to 07/13/02   21.5
Process      104136     18.72   8678.0     1.560 07/01/02 to 07/13/02   12.0
Disk          59251      3.20   2743.1     0.148 06/21/02 to 07/13/02   21.6
LVolume       85592      2.74   3962.6     0.127 06/21/02 to 07/13/02   21.6
Tran          19181      9.99    576.0     0.300 06/10/02 to 07/13/02   33.3
Overhead                 2.74
            ------- ---------  ------- ---------
TOTAL        392238     56.45  19813.9     2.636

The Global      file is now  95.4% full with room for   7.9 more full days
The Application file is now  95.2% full with room for   1.1 more full days
The Process     file is now  93.6% full with room for   0.8 more full days
The Device      file is now  59.4% full with room for  14.8 more full days
The Transaction file is now  99.9% full with room for   0.0 more full days
****************************************************************
```

PerfView

- /opt/perf/bin/pv
- View performance graphs



Data Repository Server (rep_server)

Performance Location Broker Daemon (perflbd)

Data Repository

```
    /var/opt/perf/datafiles
        logglob
        logappl
        logproc
        logdev
        logtran
        logindx
```

- Can use alternate data sources

Goals

- Ensure enough performance data exists to troubleshoot past issues
- Summarize weekly data to spot tends and facilitate capacity planning
- Compare server utilization enterprise wide to possibly support server consolidation
- Need concise per server per week reporting

Plan

1. Automate weekly performance data extract from all servers

2. Automate weekly summary data reports

3. Automate graphing of weekly summary data per server

Result

One 8 x 11 sheet

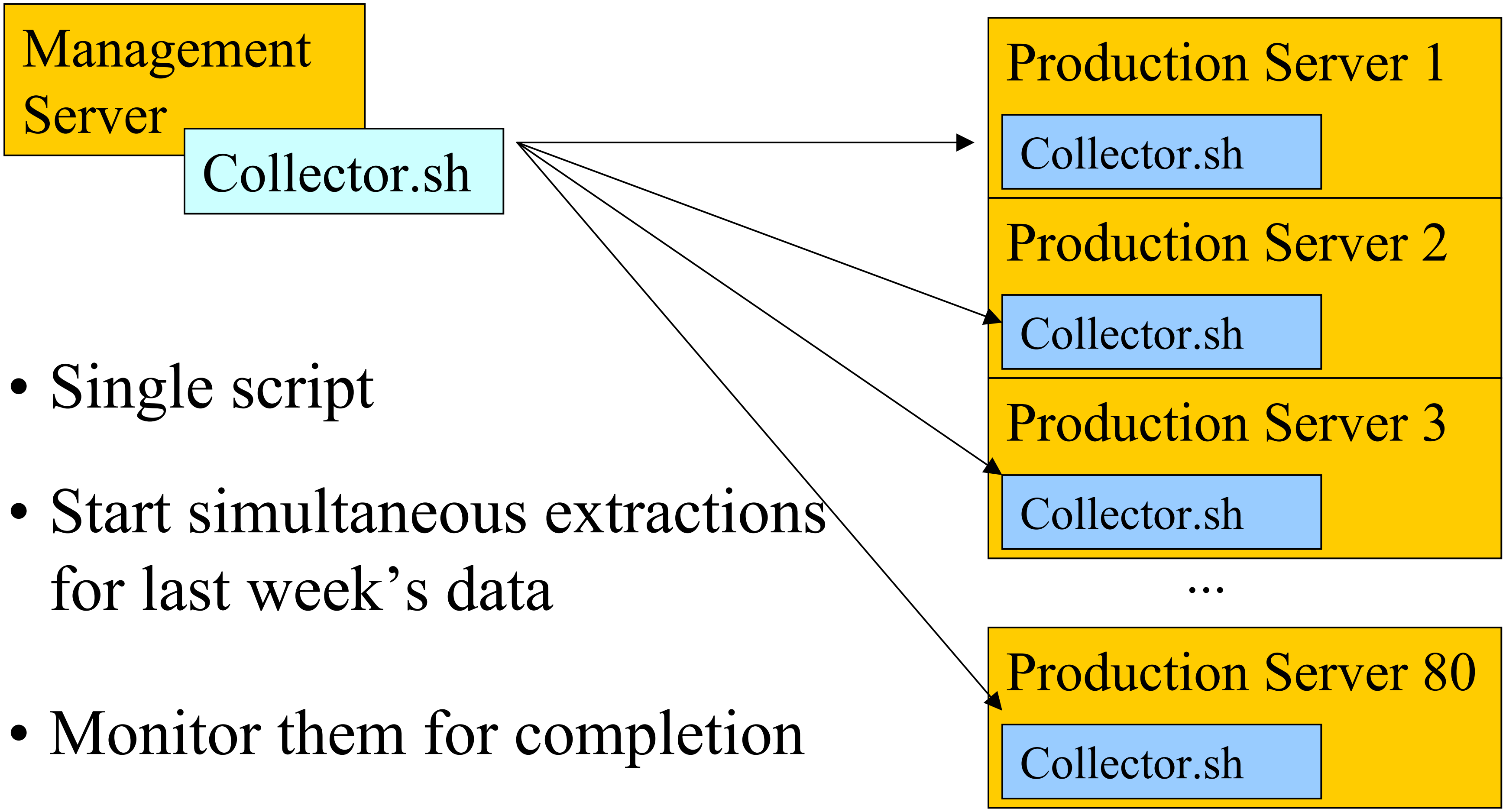
per server per week

Showing overall

CPU, Memory, I/O and LAN

Usage

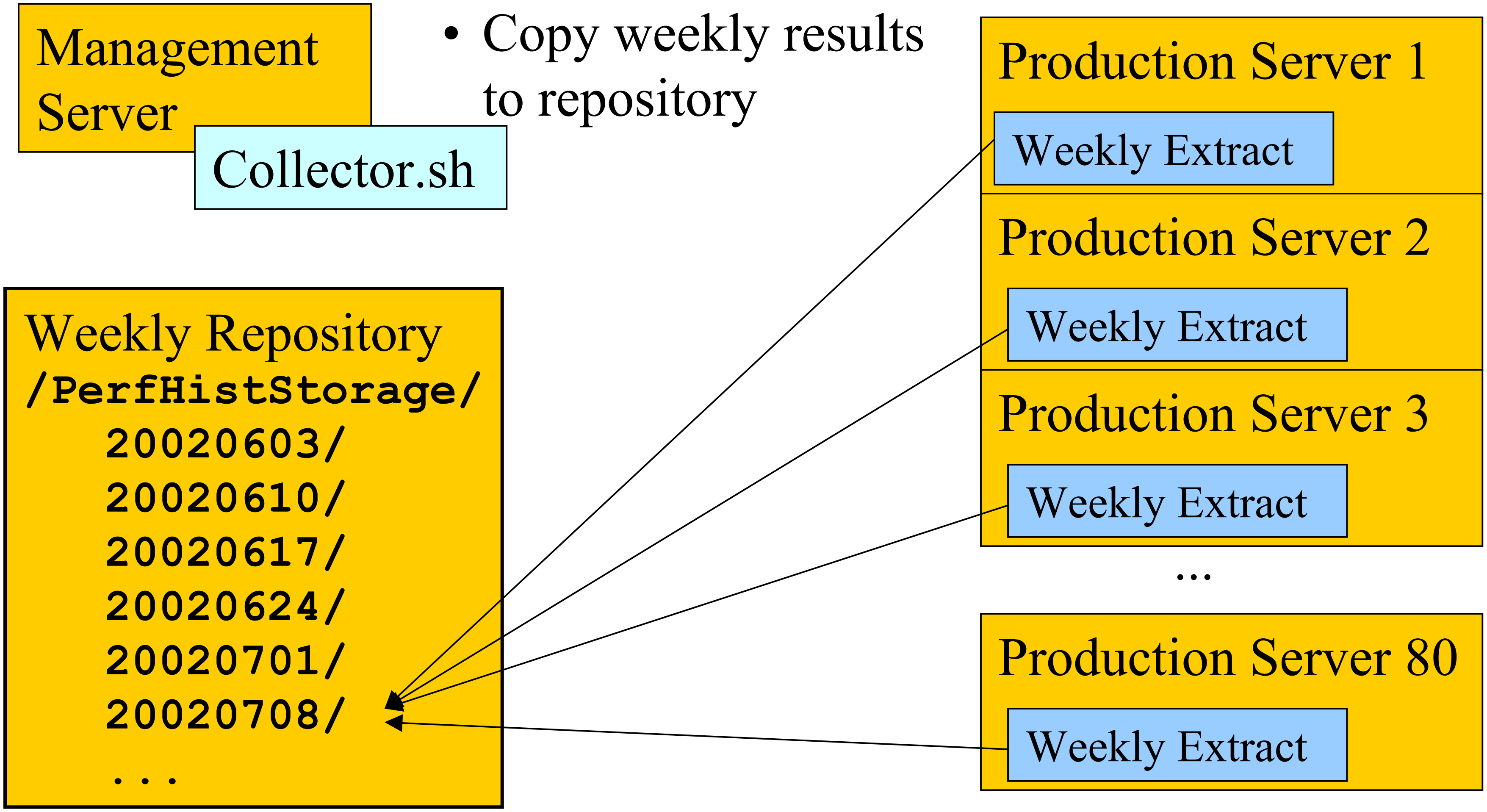
1. Automate Weekly Collection



- Single script
- Start simultaneous extractions for last week's data
- Monitor them for completion

1. Automate Weekly Collection

- Copy weekly results to repository



Management Server

Collector.sh

Weekly Repository

```
/PerfHistStorage/
   20020603/
   20020610/
   20020617/
   20020624/
   20020701/
   20020708/
   ...
```

Production Server 1

Weekly Extract

Production Server 2

Weekly Extract

Production Server 3

Weekly Extract

...

Production Server 80

Weekly Extract

Extracting History Data

```
/opt/perf/bin/extract -xt <extract options>
     -b <begin_date> -e <end_date>  -f <outputfile> -l <source>
```

- Default <source> is /opt/perf/bin/datafiles/logglob
- <extract options> are:

Summary Data	**Detail Data**
G = global	g = global
A = application	a = application
D = disk device	d = disk device
Z = logical volume	z = logical volume
N = netif	n = netif
T = transaction	t = transaction
	p = process
	k = process (killed only)
	c = configuration

- summary = 1 hour
- detail = 5 minutes (1 minute for process data)

Extracting History Data

```
/opt/perf/bin/extract -xt <extract options>
     -b <begin_date> -e <end_date> -f <outputfile> -l <source>
```

- The resulting <outputfile> can be used by PerfView and again by extract.

- <begin_date> and <end_date> formats are:

```
MM/DD/YY HH:MM AM(or PM)   (Default)
Keywords: TODAY, FIRST or LAST
<keyword>-n or <keyword>+n where n is days
```

- There are other options. See extract(1).

Extracting History Data

Example:

```
# extract -xt -gdz -b 07/08/02 00:00 \
  -e 07/14/02 23:59 -f 20020708-20020714.perf
```

- Extract detailed records from the measureware logs
- Include global, disk and logical volume metrics
- From: midnight Monday morning, July 8, 2002
-     To: midnight Sunday night, July 14, 2002.
- Data is stored in the file: 20020708-20020714.perf

Collector.sh   Server Logic

Get begin and end dates for extract

```
getweeklydates() {   # Sets $begindate and $enddate
                     # enddate is the previous Sunday at 23:59.
                     # begindate is the Monday before that at 00:00.
  let d=$(date '+%u')                     # week day (1=Monday)
  enddate="$(getdate $d) 23:59"           # get preceeding Sunday
  begindate="$(getdate $((d+=6))) 00:00"  # get Monday before that
}
getdate() {   # Return date $1 days ago from today.
 # get current month, day and year
 let month=$(date '+%m');let day=$(date '+%d');let year=$(date '+%Y')
 # subtract $1 days from current date
 ((newday=day-$1))
 while [[ $newday -lt 1 ]]; do    # If negative, subtract from month
   if ! ((month-=1)); then month=12; ((year-=1)); fi  # Subtract year
   let maxday=$(maxmonthday $month $year) # number of days in month
   ((newday+=maxday))
 done
 echo $month $newday $year | awk '{printf("%02d/%02d/%04d\n",$1,$2,$3)}'
}
```

Collector.sh Server Logic

Get begin and end dates for extract (Continued)

```
maxmonthday()  # Usage: maxmonthday <m> <y>
{
  # echo number of days in the given month and year.
  case $1 in
    '4'|'6'|'9'|'11') echo "30";;        # Apr, Jun, Sep, Nov
    '2')                                 # Feb
       if ! (($2%400)); then echo "29"   # Divisible by 400, leap year
       elif ! (($2%100)); then echo "28" # Divisible by 100, no leap year
       elif ! (($2%4)); then echo "29"   # Divisible by 4, leap year
       else echo "28"                    # else, no leap year
       fi;;
    *) echo "31";;       # Jan, Mar, May, Jul, Aug, Oct, Dec
  esac
}
```

Storage Directory

MM/DD/YYYY becomes YYYYMMDD

```
awk '{ f=split($1,a,"/"); print a[3] a[1] a[2]}'
```

Collector.sh Server Logic

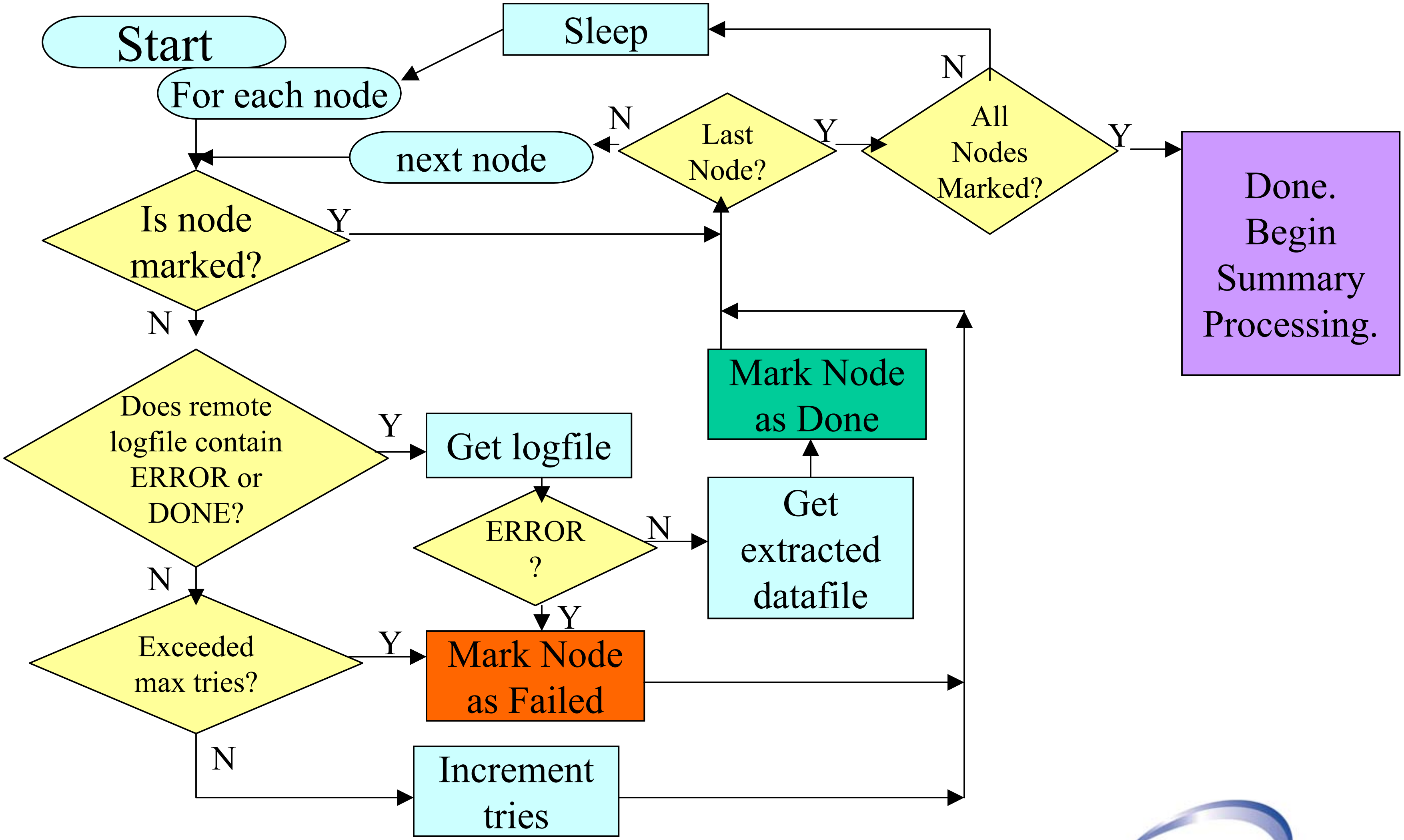
Kick off remote extracts

```
# logfile is common on all nodes
logfile="${LogDir}/${procedure}.$(date '+%Y%m%d%H%M%S').log"

# Build array of all nodes from the file $HostNames
set -A nodes $(grep -v "#" $HostNames)
let n=${#nodes[@]}-1                   # number of nodes (zero-based)

# Should the client compress the outputfile?  Command line option -c
cprs=""; cprsext=""
if [[ "$CompressOpt" = "Y" ]]; then cprs="-c"; cprsext=".Z"; fi

for node in ${nodes[@]}; do
  cmdline="remsh ${node} -n $fullprocedure -xt $DefaultExtOpts \
           -w $cprs > $logfile 2>&1 &"
  print "CMDLINE: $cmdline"
  $cmdline
done
```

```
fullprocedure = /usr/local/adm/Collector.sh  -xt option means extract
DefaultExtOpts = -gapdzcntGADZNT             -w option means weekly
```

Collector.sh   Server Logic

Monitor remote extraction jobs



Collector.sh Client Logic

Extract last week's data (each node)

```
/usr/local/adm/Collector.sh -xt -gapdzcntGADZNT -w [-c] \
      > $logfile 2>&1 &"
```

```
-w causes Collector.sh (client) to calculate:
    end_date = last Sunday night 23:59
    begin_date = Monday morning before that Sunday 00:00
```

```
# Using parameter file varaibles, locate space to extract
((found=0))
for extract_dir in $Extract_Dirs; do
 [[ -d $extract_dir ]] && let fs=$(df -k $extract_dir | \
                        awk '/free allocated/ {print $1}')
 if [[ $fs -ge $FreeSpace ]]; then ((found=1)); break; fi
done
if [[ $found -eq 0 ]]; then
  errexit "Cannot find $FreeSpace in $Extract_Dirs"
fi
bfile=$(printdatetime "${begindate}")
efile=$(printdatetime "${enddate}")
outputfile=${extract_dir}/Perf_${bfile}_${efile}.ext
```

Collector.sh Client Logic

Extract last week's data (each node) (Continued)

```
/usr/local/adm/Collector.sh -xt -gapdzcntGADZNT -w [-c] \
      > $logfile 2>&1 &"
```

```
/opt/perf/bin/extract -xt -gapdzcntGADZNT -f <outputfile> \
      -b "mm/dd/yyyy HH:MM" -e "mm/dd/yyyy HH:MM"
```

```
Example:
    -b "07/08/2002 00:00"
    -e "07/14/2002 23:59"

    -f /var/tmp/ Perf_200207080000_200207142359.ext
```

```
Communicate to Collector.sh (Server) with output:
    "DONE" = Finished
    "ERROR" = Failure
    "OUTPUTFILE" = name of extracted data file
```

```
If compress option, -c, then compress file.
```

Plan

✓ 1. Automate weekly performance data extract from all servers

   Files are:
   <hostname>_Perf_yyyymmddHHMM_yyyymmddHHMM.ext
   can be used by PerfView or extract (for exporting)

2. Automate weekly summary data reports

3. Automate graphing of weekly summary data per server

2. Automate Weekly Summary Reports

- Use Collector.sh (server side)
- For each weekly extracted file
- Use extract's export function to produce delimited ASCII data of key performancemetrics

Exporting Text Data

```
/opt/perf/bin/extract -xp  -b <begin> -e <end> -r <reportfile>
          -f <outputfile> -l <source>
```

- Default <source> is /opt/perf/bin/datafiles/logglob
- <begin> and <end> formats are:

```
MM/DD/YY HH:MM AM(or PM)   (Default)
Keywords: TODAY, FIRST or LAST
<keyword>-n or <keyword>+n where n is days
```

- <reportfile> specifies what to export into a ASCII formatted file.

Exporting Text Data

- <reportfile> specifies what to export into an ASCII file.

  /var/opt/perf/retall is a template

```
REPORT "!SYSTEM_ID !DATE !TIME !LOGFILE"
FORMAT DATAFILE      # delimited fields
HEADINGS ON
SEPARATOR="|"
SUMMARY=60           # One hour summaries
MISSING=0
DATA TYPE GLOBAL
YEAR
DAY
DATE

TIME
<performance metrics to exports>
```

Sample <reportfile>

```
REPORT "!SYSTEM_ID !DATE !TIME !LOGFILE"
FORMAT DATAFILE
HEADINGS ON
SEPARATOR="|"
SUMMARY=60
MISSING=0
DATA TYPE GLOBAL
YEAR
DAY
DATE
TIME
GBL_CPU_SYS_MODE_UTIL
GBL_CPU_USER_MODE_UTIL
GBL_MEM_USER_UTIL
GBL_MEM_SYS_AND_CACHE_UTIL
GBL_MEM_CACHE_HIT_PCT
GBL_NUM_USER
GBL_ALIVE_PROC
```

```
GBL_DISK_PHYS_IO_RATE
GBL_DISK_PHYS_IO
GBL_DISK_PHYS_BYTE
GBL_DISK_LOGL_IO_RATE
GBL_DISK_LOGL_IO
GBL_DISK_LOGL_READ_BYTE
GBL_DISK_LOGL_WRITE_BYTE
GBL_NET_PACKET_RATE
GBL_PRI_QUEUE
GBL_RUN_QUEUE
GBL_DISK_SUBSYSTEM_QUEUE
GBL_MEM_QUEUE
GBL_IPC_SUBSYSTEM_QUEUE
GBL_NETWORK_SUBSYSTEM_QUEUE
GBL_OTHER_QUEUE
```

What are Performance “metrics”?

- /opt/perf/paperdocs/mwa/C/methp.txt

  Contains a full list of performance metrics

  Global (GBL_*)

  Process (PROC_*)

  Disk (BYDISK_*)

  Network (BYNETIF_*)

  Application (APP_*)

  Transaction (TT_*)

  LVM (LV_*)

- Full descriptions, some examples, and even some indications of/for each metric are contained in methp.txt.

Examples from methp.txt (condensed)

GBL_CPU_SYS_MODE_UTIL

Percentage of time the CPU was in system mode during the interval.

A UNIX process operates in either system mode (also called kernel mode) or user mode. ...

This is NOT a measure of the amount of time used by system daemon processes, since most system daemons spend part of their time in user mode and part in system calls, like any other process.

On a system with multiple CPUs, this metric is normalized. That is, the CPU used over all processors is divided by the number of processors online. ...

High system mode CPU percentages are normal for IO intensive applications. Abnormally high system mode CPU percentages can indicate that a hardware problem is causing a high interrupt rate. It can also indicate programs that are not calling system calls efficiently.

Examples from methp.txt (condensed)

GBL_DISK_SUBSYSTEM_QUEUE

The average number of processes or threads blocked on the disk subsystem (in a "queue" waiting for their file system disk IO to complete) during the interval. This is the sum of processes or threads in the DISK, INODE, CACHE and CDFS wait states. Processes or threads doing raw IO to a disk are not included in this measurement. As this number rises, it is an indication of a disk bottleneck.

This is calculated as the accumulated time that all processes or threads spent blocked on (DISK + INODE + CACHE + CDFS) divided by the interval time.

The Global QUEUE metrics, which are based on block states, represent the average number of process or thread counts, not actual queues.

...

Some of my favorite metrics

CPU:

GBL_CPU_SYS_MODE_UTIL	System + User CPU = Total CPU
GBL_CPU_USER_MODE_UTIL	Can calculate System/User CPU ratio
GBL_ACTIVE_PROC	# of processes that used CPU / interval
~~GBL_RUN_QUEUE~~	Includes processes waiting on “short disk wait”
GBL_PRI_QUEUE	Average # of processes waiting for their priority to become high enough to get CPU.

- From methp.txt (GBL_CPU_TOTAL_UTIL)

A consistently high CPU utilization can indicate a CPU bottleneck, especially when other indicators such as **GBL_RUN_QUEUE** and **GBL_ACTIVE_PROC** are also high. High CPU utilization can also occur on systems that are bottlenecked on memory, because the CPU spends more time paging and swapping.

- From methp.txt (GBL_RUN_QUEUE)

Because this metric can include processes or threads which are waiting for disk IO to complete, it is not a reliable

CPU bottleneck indicator.

Some of my favorite metrics

IO:

GBL_DISK_LOGL/PHYS_IO_RATE	Logical/Physical IO's per second
GBL_DISK_LOGL_READ/WRITE_BYTE	KB read/written to buffer cache
GBL_DISK_PHYS_BYTE	KB transferred physically
GBL_DISK_SUBSYSTEM_QUEUE	Average # of procs block on IO
GBL_MEM_CACHE_HIT_PCT	Buffer Cache Efficiency

From methp.txt

```
GBL_DISK_PHYS_IO_RATE = GBL_DISK_FS_IO_RATE + GBL_DISK_VM_IO_RATE
+ GBL_DISK_SYSTEM_IO_RATE + GBL_DISK_RAW_IO_RATE
```

There are many reasons why there is not a direct correlation between logical IOs and physical IOs. For example, small logical writes may end up entirely in the buffer cache, and later generate fewer physical IOs when written to disk due to the larger IO size. Or conversely, small logical writes may require physical prefetching of the corresponding disk blocks before the data is merged and posted to disk. Logical volume mappings, logical disk mirroring, and disk striping also tend to remove any correlation.

Some of my favorite metrics

More Queue Metrics:

`GBL_MEM_QUEUE`

Processes blocked on memory requests.

`GBL_IPC_SUBSYSTEM_QUEUE`

Processes blocked on Inter-process Communication (IPC, MSG, SEM, PIPE, SOCKT, and STRMS wait states).

`GBL_NETWORK_SUBSYSTEM_QUEUE`

Processes blocked on Network activity (LAN, NFS, and RPC wait states).

Collector.sh Server Logic

After remote extraction jobs are done

Summary Processing

```
/usr/local/adm/ExportPerf.sh -i <inputdir> -r <reportfile>
```

For each file <inputdir>/*.ext execute:

```
extract -xp -b FIRST -e LAST -l <file> -r <reportfile> -f <outputfile>
```

Produces: **`<host>_Perf_yyyymmddHHMM_yyyymmddHHMM.exp`**

Plan

✓ 1. Automate weekly performance data extract from all servers

✓ 2. Automate weekly summary data reports

3. Automate graphing of weekly summary data per server

3. Automate Graphs Using Excel Macros

- Imported summary data into sheets
- Create Global Chart
  - User and System CPU
  - User, System and Cache Memory Usage
  - Cache hit rate
- Create I/O Chart
  - Logical Read/Write GB/hour
  - Physical GB/hour
- Create Global Queue Chart
  - Priority, Run, Disk Subsystem, Memory and Network Queues
- Create LAN Usage Chart
  - packet rate

Excel Visual Basic (main)

```
FilesToOpen = Application.GetOpenFilename("Perf Files (*.exp), *.exp", , , , True)
If IsArray(FilesToOpen) Then
    docharts = MsgBox("Create Charts?", vbYesNoCancel)
    If docharts = vbCancel Then
        Exit Sub
    End If
    ' The number of files is determined by the filesToOpen array
    numberoffiles = UBound(FilesToOpen)

    ' If charts are created, each file becomes its own workbook with data and charts.
    ' If charts are not to be created, a single workbook holds all of the data sheets.
    If numberoffiles > 1 And docharts = vbNo Then
         Workbooks.Add                                  ' Create a Single Workbook
         n = ActiveWorkbook.Sheets.Count                ' These initial sheets will
         workbookname = ActiveWorkbook.Name             ' be deleted later.
    End If
```

Excel Visual Basic

```
' Master loop. For each selected file...
 For i = 1 To numberoffiles
    fileToOpen = FilesToOpen(i)
    ' Open and format the data file. Get the sheet name.
    Call OpenPerfFile(fileToOpen, sheetname)
    ' If multiple files and no charts, move to single workbook.
    If numberoffiles > 1 And docharts = vbNo Then
            Sheets(sheetname).Move After:=Workbooks(workbookname).Sheets(n)
    End If
    If docharts = vbYes Then                              ' Begin creating charts
      ' Create Global Chart
      Call CreatePerfChart(sheetname, "E1:J170", 6, "Global", "Global Performance", _
                                      "Percent", True, False)
      ' Create Queue Chart
      Call CreatePerfChart(sheetname, "U1:X170,Z1:Z170", 5, "Queue", "Queue Depth", _
                                        "Number", False, False)
      ' Create IO Volume Chart
      Call CreatePerfChart(sheetname, "O1:O170,R1:S170", 3, "IO Volume", _
                                        "GB per Hour", "GB", False, False)
      ' Create Network Chart
      Call CreatePerfChart(sheetname, "T1:T170", 1, "Network", "Network Performance",_
                                        "Packet Rate", False, False)
```

Excel Visual Basic

```
Sub OpenPerfFile(PerfFileName, PerfSheetName)
  Workbooks.OpenText FileName:=PerfFileName, Origin:=xlWindows, StartRow:=2,
    DataType:=xlDelimited, TextQualifier:=xlDoubleQuote, ConsecutiveDelimiter:=False,
    Tab:=False, Semicolon:=False, Comma:=False, Space:=False, Other:=True,
    OtherChar:="|", FieldInfo:=Array(Array(1, 1), Array(2, 1), Array(3, 3), Array(4, 1), _
    Array(5, 1), Array(6, 1), Array(7, 1), Array(8, 1), Array(9, 1), Array(10, 1), Array(11, 1),_
    Array(12, 1), Array(13, 1), Array(14, 1), Array(15, 1), Array(16, 1), Array(17, 1), _
    Array(18, 1), Array(19, 1), Array(20, 1), Array(21, 1), Array(22, 1), Array(23, 1), _
    Array(24, 1), Array(25, 1), Array(26, 1), Array(27, 1))
  PerfSheetName = ActiveSheet.Name                  ' Return the name of the sheet
  ' Insert column for Total CPU% as System CPU% + User CPU%
  range("G1").EntireColumn.Insert                             ' Insert column
  range("G1").FormulaR1C1 = "Total"                           ' Column header 1
  range("G2").FormulaR1C1 = "CPU%"                            ' Column header 2
  range("G3:G170").FormulaR1C1 = "=RC[-2]+RC[-1]"      ' Formula for total CPU%
 ' Convert KB to GB for "Phys KB", "Logl Rd KB", and "Logl Wr KB"
 Call ConvertBytes(3, 15, 1048576, "0.000")
 Cells(2, 15) = "GB"   ' Change header
 Call ConvertBytes(3, 18, 1048576, "0.000")
 Cells(2, 18) = "Rd GB"   ' Change header
 Call ConvertBytes(3, 19, 1048576, "0.000")
 Cells(2, 19) = "Wr GB"   ' Change header
End Sub
```

Excel Visual Basic

```
Sub CreatePerfChart(sheetname, DataRange As String, _
            nranges As Integer, chartname As String, _
            ctitle As String, atitle As String, percent As Boolean, _
            blackwhite As Boolean)
 ' sheetname   Name of sheet with data to chart
 ' DataRange   Range specification of data to chart
 ' nranges     Number of columns of data to chart
 ' chartname   Name to apply to the chart sheet
 ' ctitle      Title of the chart. Will be sheetname & chr$(130 & ctitle
 ' atitle      Title for the Y-axis
 ' percent     Is Y-axis as percentage scale?
 ' blackwhite  Should chart be Black & White?
 Dim x As Integer
 Charts.Add
 ActiveChart.SetSourceData Source:=Sheets(sheetname).range(DataRange), _
    PlotBy:=xlColumns

 For x = 1 To nranges
  ActiveChart.SeriesCollection(x).XValues = "=" & sheetname & "!R3C3:R170C4"
 Next x
```

Excel Visual Basic

Sub CreatePerfChart (continued)

```
With ActiveChart
   .ChartType = xlLine
   .Location Where:=xlLocationAsNewSheet, Name:=chartname
   .HasTitle = True
   .ChartTitle.Characters.text = sheetname & Chr$(13) & ctitle
   .Axes(xlCategory, xlPrimary).HasTitle = False
   .Axes(xlValue, xlPrimary).HasTitle = True
   .Axes(xlValue, xlPrimary).AxisTitle.Characters.text = atitle
   .Axes(xlCategory).HasMajorGridlines = True
   .Axes(xlCategory).HasMinorGridlines = False
   .Axes(xlValue).HasMajorGridlines = True
   .Axes(xlValue).HasMinorGridlines = False
   .HasDataTable = False
   .Axes(xlCategory).CrossesAt = 1
   .Axes(xlCategory).TickLabelSpacing = 12
   .Axes(xlCategory).TickMarkSpacing = 12
   .Axes(xlCategory).AxisBetweenCategories = True
   .PlotArea.Border.Weight = xlThin
   .PlotArea.Border.LineStyle = xlNone
   .PlotArea.Interior.ColorIndex = xlNone
 End With
```

Excel Visual Basic

Sub CreatePerfChart (continued)

```
 If percent Then                            ' percent chart has 100 scale
    With ActiveChart.Axes(xlValue)
      .MinimumScaleIsAuto = True
      .MaximumScale = 100
      .MinorUnitIsAuto = True
      .MajorUnitIsAuto = True
      .Crosses = xlAutomatic
      .ReversePlotOrder = False
      .ScaleType = xlLinear
    End With
  End If
  If blackwhite Then
       ActiveChart.PageSetup.BlackAndWhite = True
  End If
   If nranges = 1 Then                      ' Only 1 data series, no legend
      ActiveChart.legend.Delete
   Else
      ActiveChart.legend.Position = xlLegendPositionTop
   End If
End Sub
```

Excel Visual Basic

```
Sub ConvertBytes(row As Integer, col As Integer, div As Long, format As String)
   ' Divide column (col) of data from row (row) to row 170 by (div) and apply
   ' number format (format).
   Dim col1 As Integer
   col1 = col + 1

   ' Insert column after given column.
   range(Cells(row, col1).Address).EntireColumn.Insert
   ' Place formula into new column.
   range(Cells(row, col1).Address + ":" + Cells(170, col1).Address).FormulaR1C1 = _
       "=RC[-1]/" + Str(div)
   ' Copy new data (values only) on top of original column
   range(Cells(row, col1).Address + ":" + Cells(170, col1).Address).Copy
   range(Cells(row, col).Address).Select
   Selection.PasteSpecial Paste:=xlValues, Operation:=xlNone, SkipBlanks:= _
        False, Transpose:=False
   range(Cells(row, col1).Address).EntireColumn.Delete   ' Delete inserted column
   ' Apply number format (format)
   range(Cells(row, col).Address + ":" + Cells(170, col).Address).NumberFormat = _
              format
End Sub
```

Excel Visual Basic

main continued after charts created

```
    ' create summary chart sheet
    Sheets.Add
    ActiveSheet.Name = "Summary"
    summarysheet = "Summary"
    With ActiveSheet.PageSetup
      .LeftMargin = Application.InchesToPoints(0.3)
      .RightMargin = Application.InchesToPoints(0.3)
      .TopMargin = Application.InchesToPoints(0.5)
      .BottomMargin = Application.InchesToPoints(0.5)
      '.HeaderMargin = Application.InchesToPoints()
      '.FooterMargin = Application.InchesToPoints()
      .Zoom = False
      .FitToPagesWide = 1
      .FitToPagesTall = 1
    End With
    Call PasteChart("Global", summarysheet, 1, True, False)
    Call PasteChart("Queue", summarysheet, 2, True, True)
    Call PasteChart("IO Volume", summarysheet, 3, True, True)
    Call PasteChart("Network", summarysheet, 4, False, True)
  End If              ' End if to do charts?
Next i                ' Next Summary Performance File
```

Excel Visual Basic

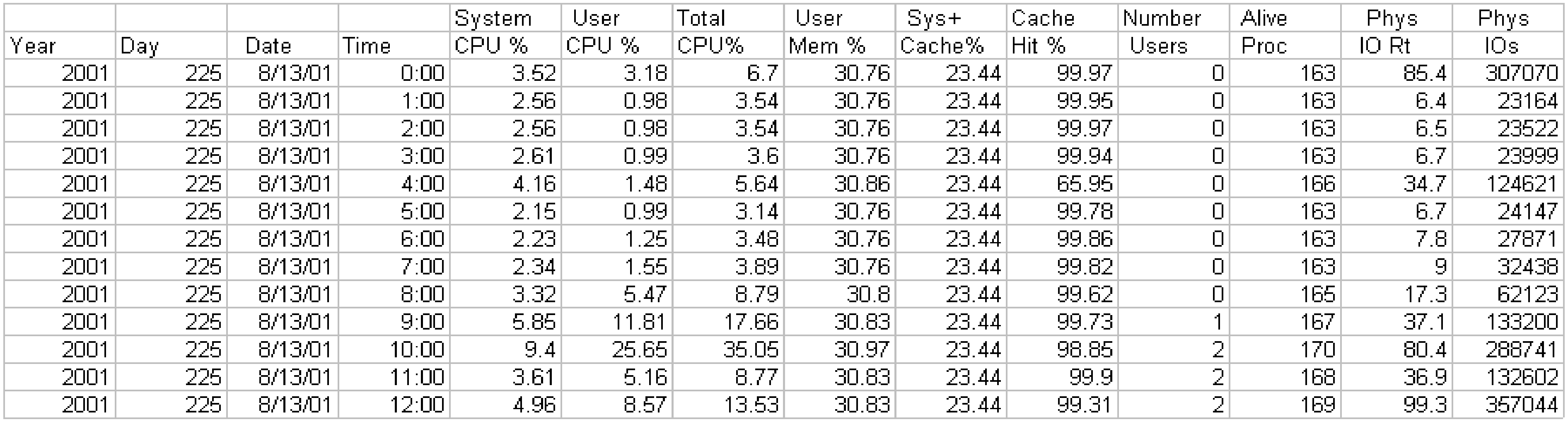
```
Sub PasteChart(chart As String, dest, chartnum As Integer, legend As Boolean, removeX)
   ' chart         Sheetname of chart sheet (source)
   ' dest          Sheetname of destinate sheet
   ' chartnum      Chart number on sheet (1=Top)
   ' legend        Chart has a legend?
   ' removeX       Remove X axis labels?
 GHeight = 170                      ' Chart height
 GWidth = 500                       ' Chart width
 chartname = "Chart " + Str(chartnum)
 Sheets(chart).ChartArea.Copy
 Sheets(dest).Select
 ActiveSheet.Paste
 With ActiveSheet.Shapes(chartname)
  .Left = 1
  .Top = 1 + (chartnum - 1) * GHeight
  .Height = GHeight
  .Width = GWidth
 End With
 ActiveSheet.ChartObjects(chartname).Activate
 If legend Then
  ActiveChart.legend.Position = xlTop
 End If
 If removeX Then
  ActiveChart.Axes(xlCategory).Delete
 End If
End Sub
```

Excel Visual Basic

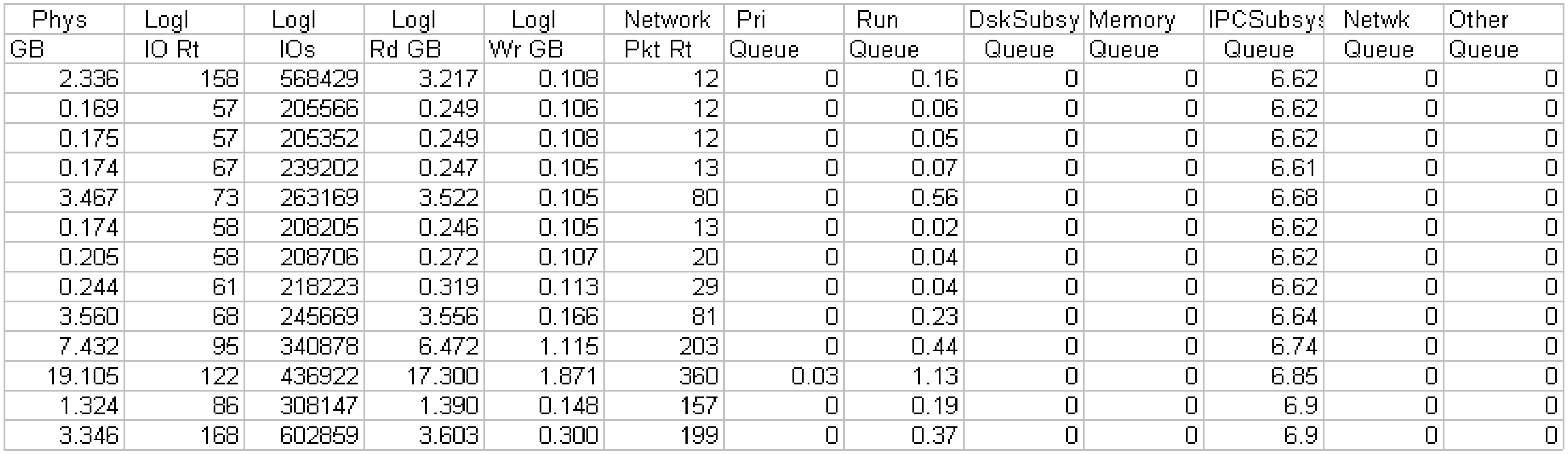
main finished

```
 ' Delete original sheets
 If numberoffiles > 1 And docharts = vbNo Then
  Sheets(1).Select
  For j = 2 To n
   Sheets(j).Select (False)
  Next j
  Application.DisplayAlerts = False
  ActiveWindow.SelectedSheets.Delete
 End If
End If     ' End if files to open
```

Samples

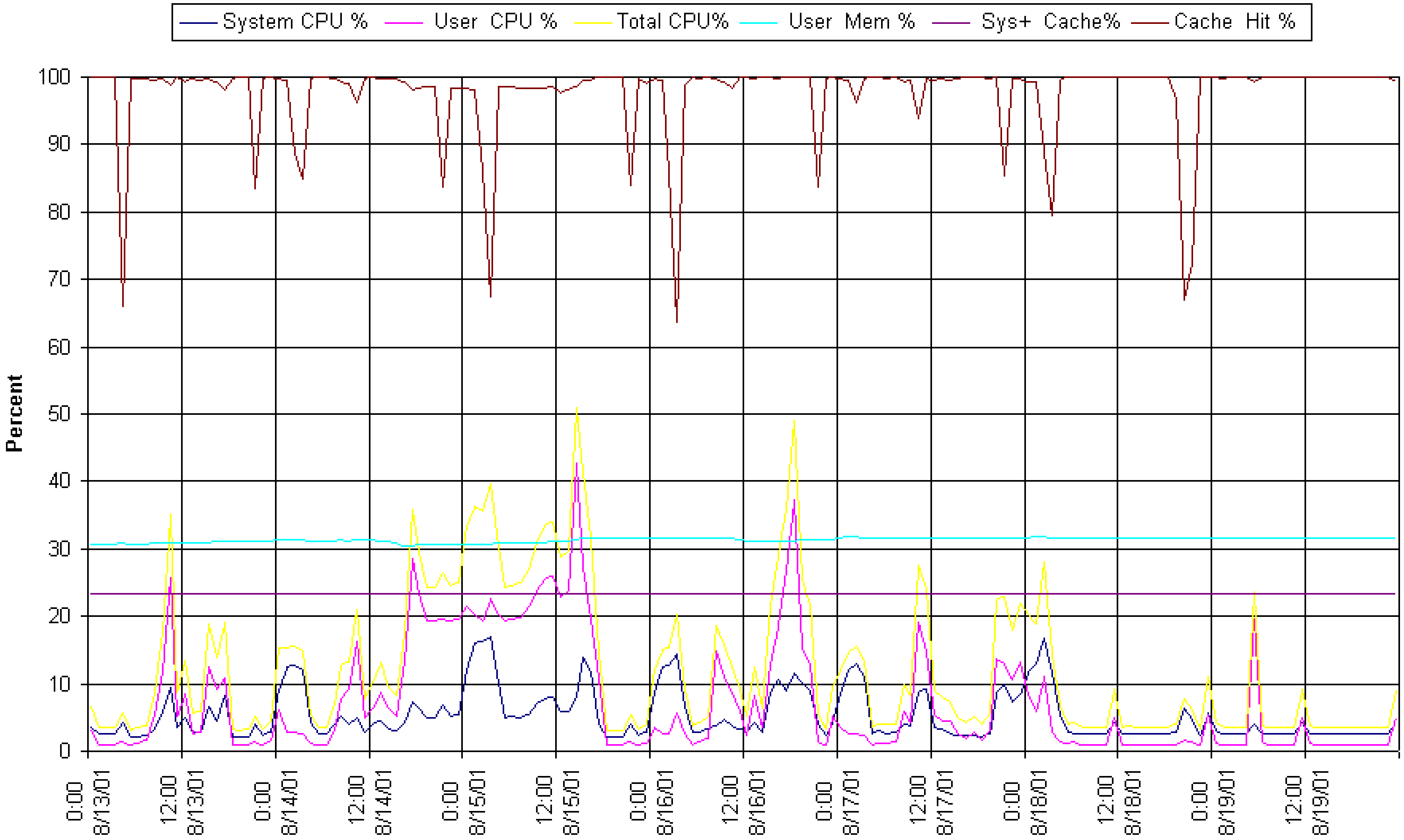


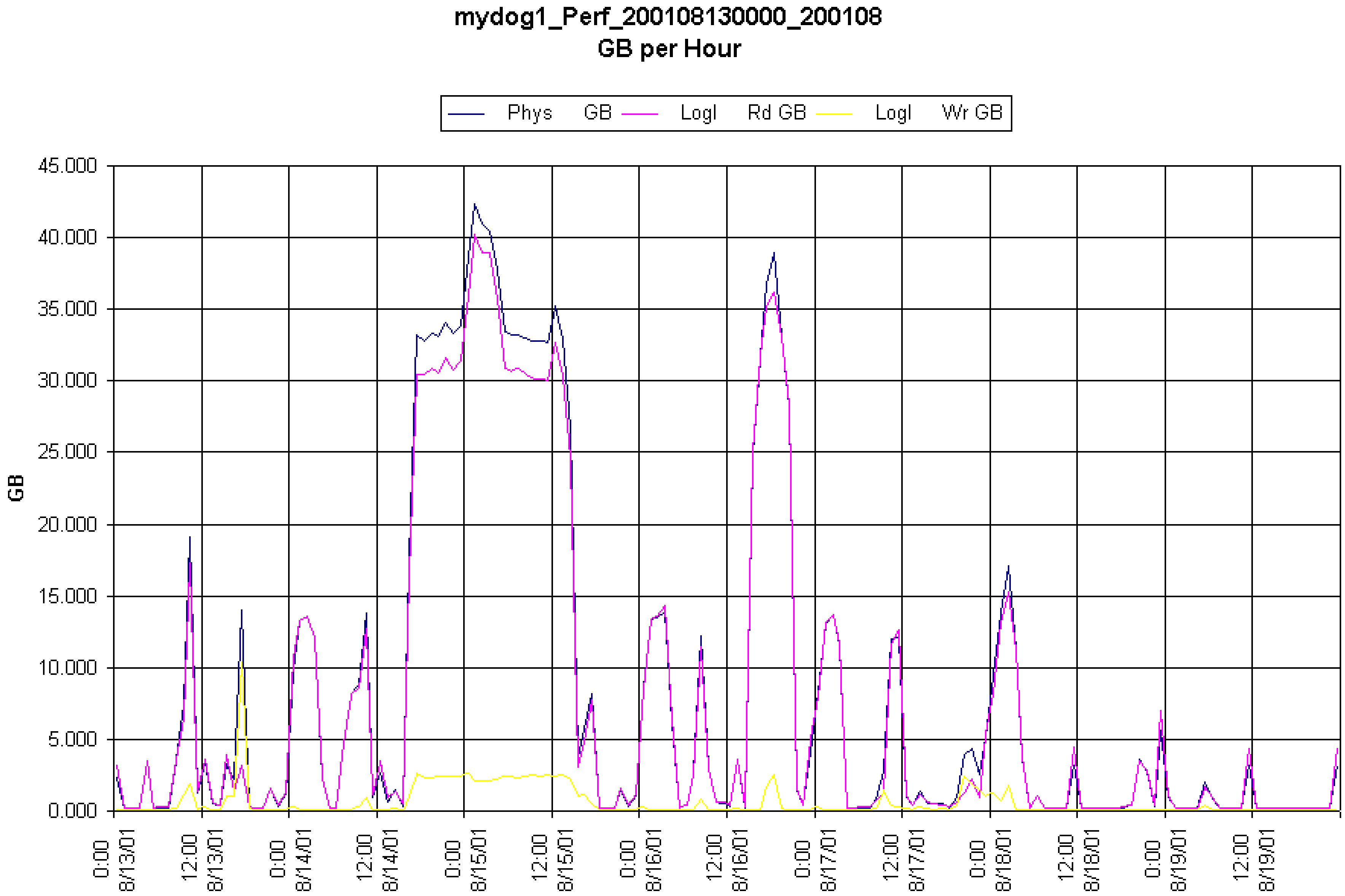
Year	Day	Date	Time	System CPU %	User CPU %	Total CPU%	User Mem %	Sys+ Cache%	Cache Hit %	Number Users	Alive Proc	Phys IO Rt	Phys IOs
2001	225	8/13/01	0:00	3.52	3.18	6.7	30.76	23.44	99.97	0	163	85.4	307070
2001	225	8/13/01	1:00	2.56	0.98	3.54	30.76	23.44	99.95	0	163	6.4	23164
2001	225	8/13/01	2:00	2.56	0.98	3.54	30.76	23.44	99.97	0	163	6.5	23522
2001	225	8/13/01	3:00	2.61	0.99	3.6	30.76	23.44	99.94	0	163	6.7	23999
2001	225	8/13/01	4:00	4.16	1.48	5.64	30.86	23.44	65.95	0	166	34.7	124621
2001	225	8/13/01	5:00	2.15	0.99	3.14	30.76	23.44	99.78	0	163	6.7	24147
2001	225	8/13/01	6:00	2.23	1.25	3.48	30.76	23.44	99.86	0	163	7.8	27871
2001	225	8/13/01	7:00	2.34	1.55	3.89	30.76	23.44	99.82	0	163	9	32438
2001	225	8/13/01	8:00	3.32	5.47	8.79	30.8	23.44	99.62	0	165	17.3	62123
2001	225	8/13/01	9:00	5.85	11.81	17.66	30.83	23.44	99.73	1	167	37.1	133200
2001	225	8/13/01	10:00	9.4	25.65	35.05	30.97	23.44	98.85	2	170	80.4	288741
2001	225	8/13/01	11:00	3.61	5.16	8.77	30.83	23.44	99.9	2	168	36.9	132602
2001	225	8/13/01	12:00	4.96	8.57	13.53	30.83	23.44	99.31	2	169	99.3	357044

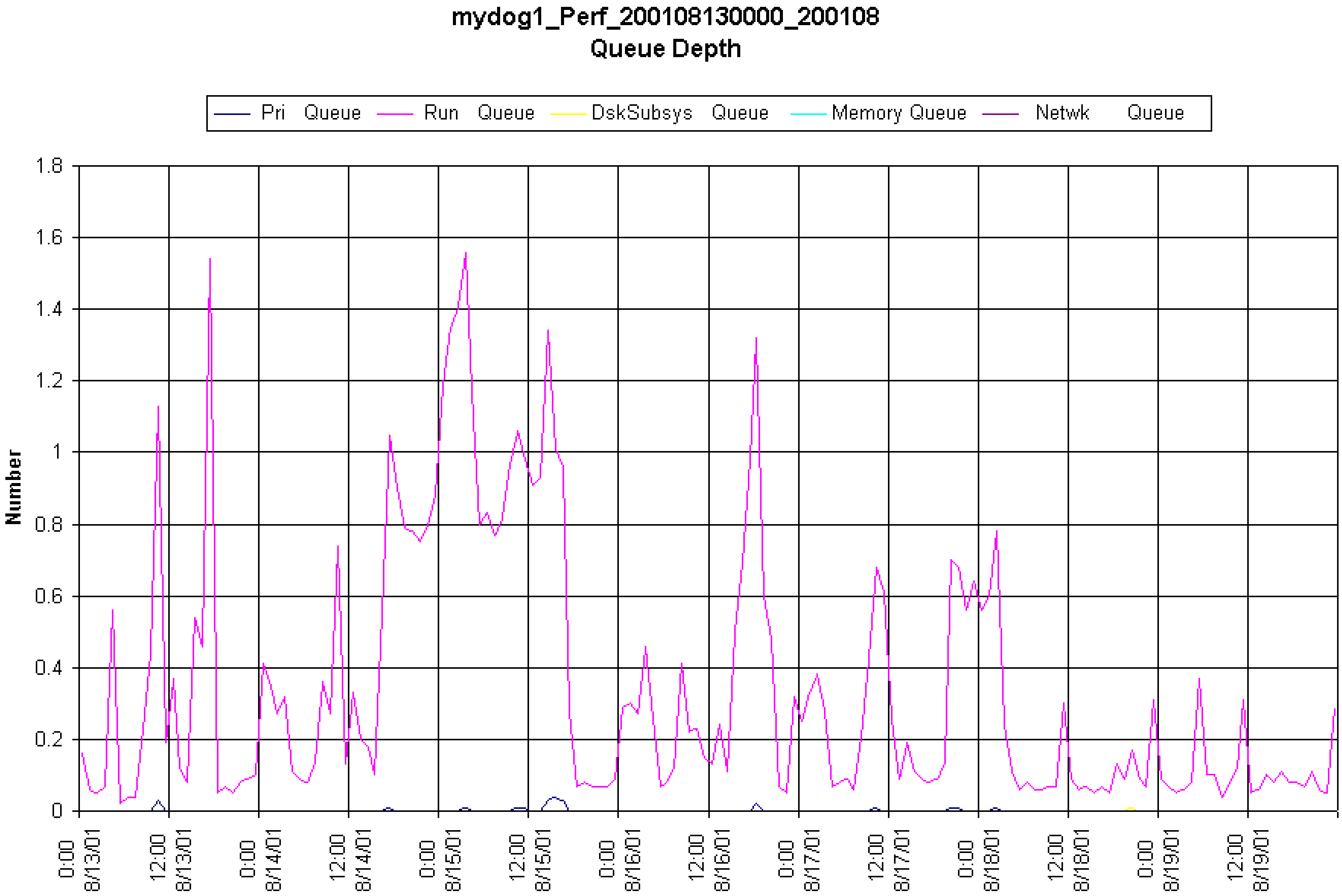


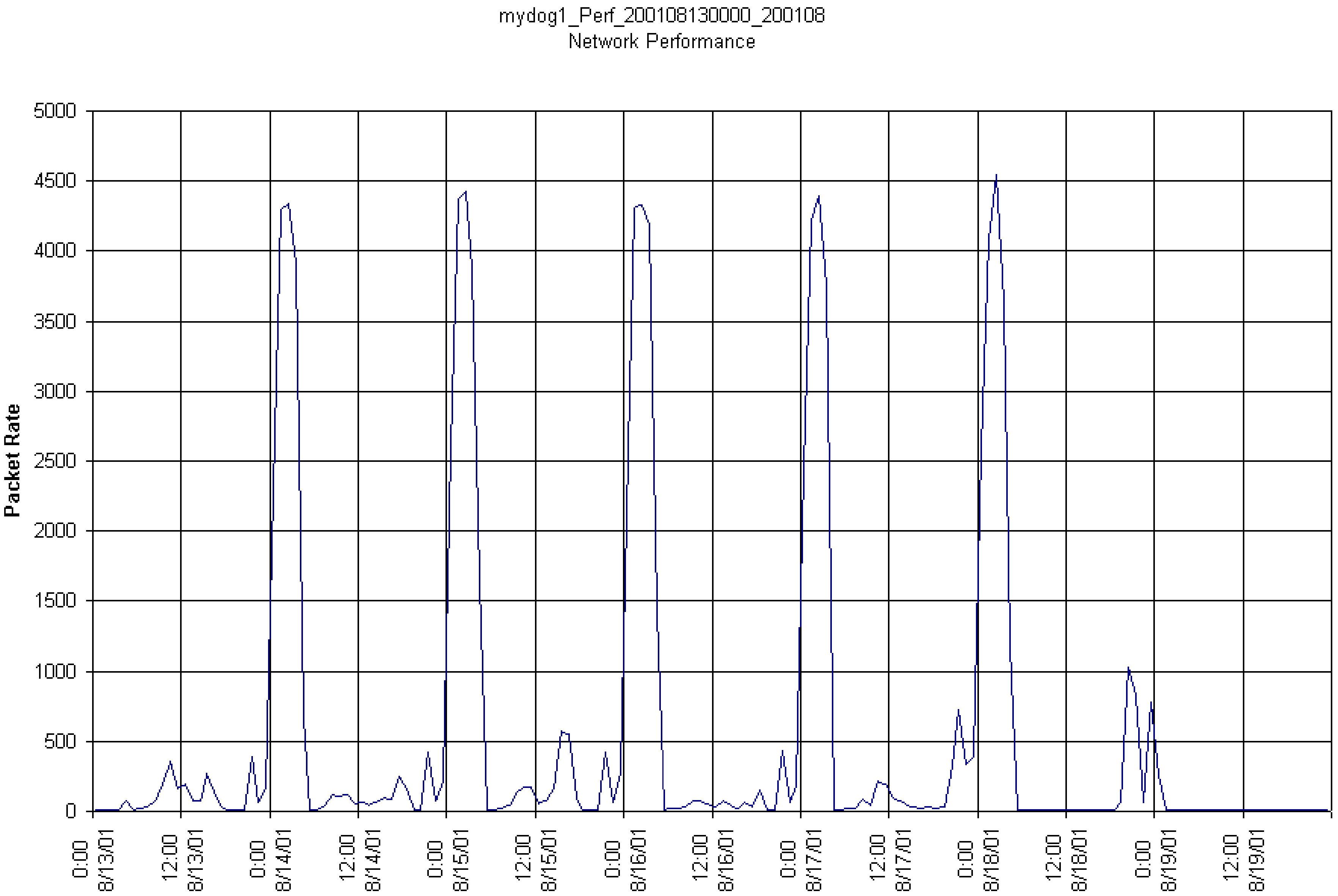
Phys GB	Logl IO Rt	Logl IOs	Logl Rd GB	Logl Wr GB	Network Pkt Rt	Pri Queue	Run Queue	DskSubsy Queue	Memory Queue	IPCSubsys Queue	Netwk Queue	Other Queue
2.336	158	568429	3.217	0.108	12	0	0.16	0	0	6.62	0	0
0.169	57	205566	0.249	0.106	12	0	0.06	0	0	6.62	0	0
0.175	57	205352	0.249	0.108	12	0	0.05	0	0	6.62	0	0
0.174	67	239202	0.247	0.105	13	0	0.07	0	0	6.61	0	0
3.467	73	263169	3.522	0.105	80	0	0.56	0	0	6.68	0	0
0.174	58	208205	0.246	0.105	13	0	0.02	0	0	6.62	0	0
0.205	58	208706	0.272	0.107	20	0	0.04	0	0	6.62	0	0
0.244	61	218223	0.319	0.113	29	0	0.04	0	0	6.62	0	0
3.560	68	245669	3.556	0.166	81	0	0.23	0	0	6.64	0	0
7.432	95	340878	6.472	1.115	203	0	0.44	0	0	6.74	0	0
19.105	122	436922	17.300	1.871	360	0.03	1.13	0	0	6.85	0	0
1.324	86	308147	1.390	0.148	157	0	0.19	0	0	6.9	0	0
3.346	168	602859	3.603	0.300	199	0	0.37	0	0	6.9	0	0

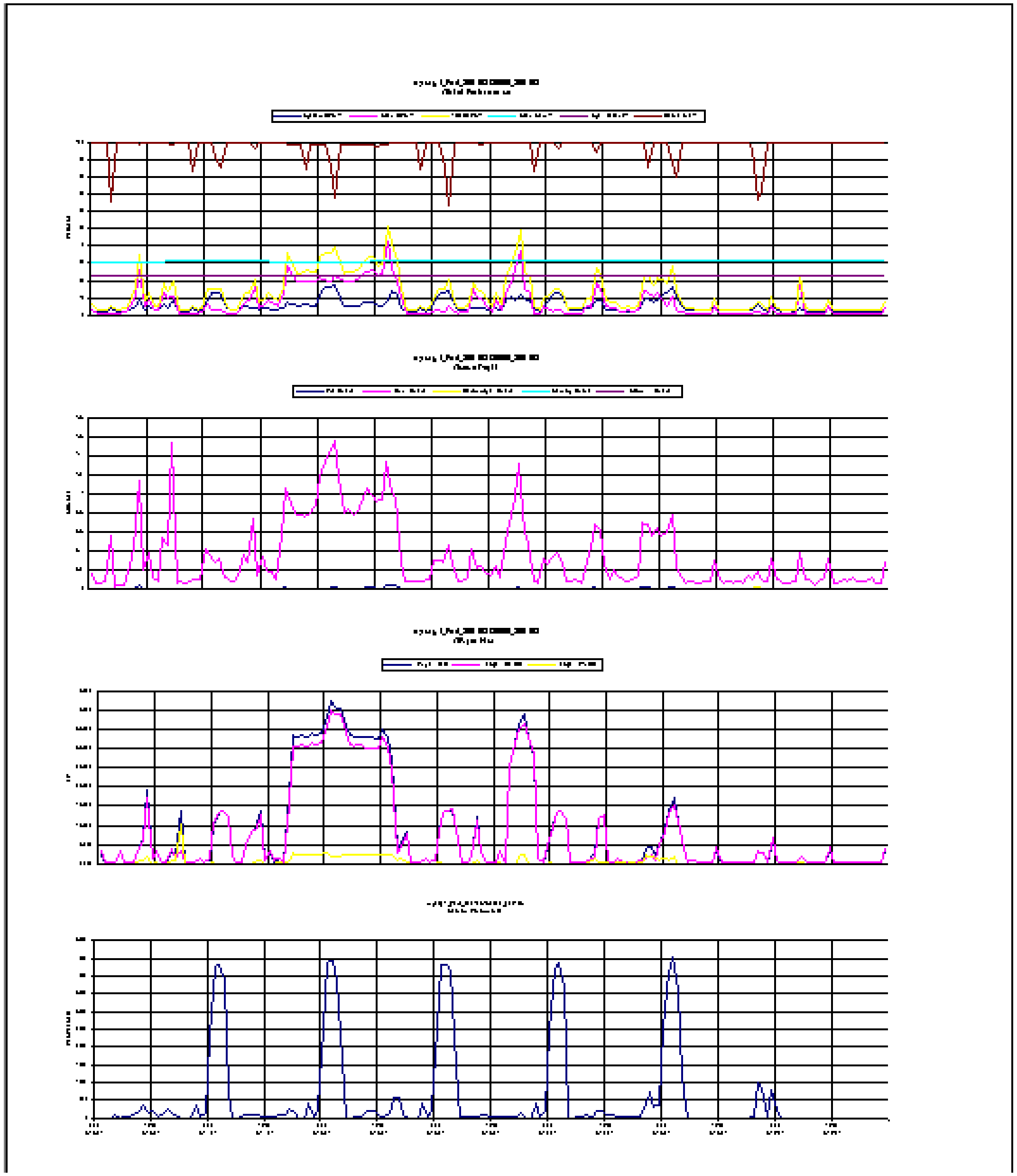
mydog1_Perf_200108130000_200108
Global Performance




mydog1_Perf_200108130000_200108
GB per Hour
Phys GB
Logl Rd GB
Logl Wr GB
GB
45.000
40.000
35.000
30.000
25.000
20.000
15.000
10.000
5.000
0.000
0:00 8/13/01
12:00 8/13/01
0:00 8/14/01
12:00 8/14/01
0:00 8/15/01
12:00 8/15/01
0:00 8/16/01
12:00 8/16/01
0:00 8/17/01
12:00 8/17/01
0:00 8/18/01
12:00 8/18/01
0:00 8/19/01
12:00 8/19/01


mydog1_Perf_200108130000_200108
Queue Depth
Pri Queue
Run Queue
DskSubsys Queue
Memory Queue
Netwk Queue
Number
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
0:00 8/13/01
12:00 8/13/01
0:00 8/14/01
12:00 8/14/01
0:00 8/15/01
12:00 8/15/01
0:00 8/16/01
12:00 8/16/01
0:00 8/17/01
12:00 8/17/01
0:00 8/18/01
12:00 8/18/01
0:00 8/19/01
12:00 8/19/01


mydog1_Perf_200108130000_200108
Network Performance
Packet Rate
5000
4500
4000
3500
3000
2500
2000
1500
1000
500
0
0:00 8/13/01
12:00 8/13/01
0:00 8/14/01
12:00 8/14/01
0:00 8/15/01
12:00 8/15/01
0:00 8/16/01
12:00 8/16/01
0:00 8/17/01
12:00 8/17/01
0:00 8/18/01
12:00 8/18/01
0:00 8/19/01
12:00 8/19/01



Result
One 8 x 11 sheet
per server per week
Showing overall
CPU, Memory, I/O and
LAN Usage