

MeasureWare and PerfView for Windows NT

Paper: 4090
Laurence Sweeney
Network & System Management Division
Resource & Performance Management
Hewlett-Packard Company

Introduction

This paper looks at OpenView MeasureWare and OpenView PerfView as implemented on Windows NT¹. The functionality of both products is reviewed and Windows NT specific features are highlighted. The paper concludes with a discussion of how these tools may be used to help manage a mixed UNIX² and Windows NT environment.

MeasureWare

MeasureWare is a truly autonomous intelligent agent, as much at home on a single node, as it is deployed across the workgroup or enterprise. The MeasureWare agent provides three key services:

- Data Collection
- Import & Export of Data through open interfaces
- Correlated Service Management alarms

Data Collection

The MeasureWare agent collects six types of data: Global, Application, Process, Device, Application Response Measurement (ARM) and Data Source Integration (DSI). MeasureWare is a low overhead collector (around 2% of System CPU), which runs permanently in the background. It uses patented technology to manage the collection process.

Global data consists of metrics, which (as the name suggests) are global across a system. Likewise process data presents metrics specific to a unique process. Application data consists of a user defined grouping of process data. By appropriate grouping of processes it is possible to get measures of application resource usage *without* having to instrument the application code directly. This specification takes place within the MeasureWare Parm file. (See Appendix A for a sample MeasureWare Parm file from an NT system)

Making use of MeasureWare Application Grouping is an extremely powerful tool for service management. Without any change to application code it is possible to get an accurate representation of application resource consumption.

¹ Microsoft Windows, Windows NT, and Windows 95 (WIN95) are trademarks of Microsoft Corporation.

² UNIX is a registered trademark of X/Open inc.

Application Response Measurement

Although MeasureWare can be used to group processes into applications in some cases this is not enough. End-to-end response time around a business transaction is typically of great importance in a Service Management environment. The Application Response Measurement API allows for the instrumentation of applications in a vendor-neutral fashion. MeasureWare is able to collect data from ARM enabled applications. For more information on ARM see <http://www.hp.com/go/arm>. See also “Managing the Enterprise with the Application Response Measurement API (ARM)”³ by Denise Morris.

The MeasureWare agent is available across a wide range of vendors equipment including HP, IBM, BULL, Microsoft, NCR, Siemens Nixdorf and Sun.⁴ All agents are configured and operate in a similar manner and return the same data (where available). They provide a consistent metric naming strategy across a heterogeneous environment. This consistent view across the environment helps cut through the confusion that often exists when trying to compare and contrast data from different platforms.

Data Import and Export

All MeasureWare agents provide what is known as Data Source Integration (DSI). This provides an open interface for the import of data into MeasureWare. DSI is designed to take a stream of ASCII data, summarize it, timestamp and log into MeasureWare logfiles. Once this data is in the MeasureWare environment it can be treated just like any other type of data. Correlated alarms can be built, PerfView can analyze the data and MeasureWare can Export it for use in other applications.

DSI has been used to implement a number of RPM solutions. For example, the MeasureWare NOS Connectivity module uses DSI to bring data from Windows NT into a MeasureWare host running on a UNIX server.

Most DSI implementations fall into two major categories: a proxy solution or an application management solution. A proxy solution might be built when there is a need to manage an operating system for which there is no native MeasureWare agent; or where there is no need for a full MeasureWare agent and a selection of Global Metrics (so-called health metrics) is all that is required. The MeasureWare NOS Connectivity Modules are an example of a proxy solution. The Network Response Facility (*nrf*) which is included with the HP-UX MeasureWare Agent, is an example of an application management solution. The *nrf* takes data from the HP Netmetrix Internetwork Response Manager (IRM) and brings it into the MeasureWare environment. The IRM gathers data about network response time between servers on a network. Integration of this kind of network performance information into MeasureWare allows for a clear view of key service management components.

Many user applications produce logfiles as part of their normal operation. These logfiles can be integrated into MeasureWare through DSI, strengthening the Service Management solution *without* having to instrument the application. Many off-the-shelf applications lend themselves to this kind of Application Management Solution.

³ Available from [//www.hp.com/go/rpm](http://www.hp.com/go/rpm)

⁴ See [//www.hp.com/go/rpm](http://www.hp.com/go/rpm) for a current list of MeasureWare agents.
MeasureWare and PerfView for Windows NT

There are two main routes for the analysis or reporting of MeasureWare Data:

- PerfView Analyzer
- MeasureWare Extract/Export.

MeasureWare is supplied with an Extract program that allows for the Export of data in a number of different formats including: Spreadsheet, ASCII, BINARY and Comma-Separated Variable (CSV). A report file is used to specify which metrics should be exported. For example consider the following which will print a report breaking down the CPU Utilization into System, User, Interrupt and Idle CPU Utilization.

```
extract -b today-7 -l C:\RPMTOOLS\data\datafiles\logglob -r myrept -g -xp
```

this command runs extract in batch, requests an export of the last seven days worth of global summary data from the file in C:\RPMTOOLS\data\datafiles\logglob using the report file called myrept.

Myrept is a report layout file and looks like this:

```
REPORT "MWA Export !DATE !TIME Logfile: !LOGFILE !COLLECTOR !SYSTEM_ID"
FORMAT ASCII
HEADINGS ON
SEPARATOR="|"
SUMMARY=60
```

DATA TYPE GLOBAL

```
DATE
TIME
GBL_CPU_TOTAL_UTIL
GBL_CPU_SYS_MODE_UTIL
GBL_CPU_USER_MODE_UTIL
GBL_CPU_INTERRUPT_UTIL
GBL_CPU_IDLE_UTIL
```

running extract as above with this report file produces an ASCII file called xfrdGLOBAL.asc. If you look at this file you will see

Export	01/27/97	SCOPE	X.00.10	50862SWE			
			System	User	Intrpt	Idle	
	Date	Time	CPU %	CPU %	Time	CPU %	CPU %
	01/24/97	12:10	0.55	0.30	0.734	0.02	99.43
	01/24/97	12:15	61.67	16.82	0.448	0.00	37.38
	01/24/97	12:15	0.72	0.14	1.770	0.02	99.25
	01/24/97	12:20	1.09	0.68	1.225	0.18	98.90
	01/24/97	12:25	1.24	0.72	1.560	0.24	98.74
	01/24/97	12:30	2.63	1.80	2.481	0.20	97.36
	01/24/97	12:35	4.91	3.94	2.910	0.44	95.07
	01/24/97	12:40	6.38	5.26	3.348	0.32	93.60
	01/24/97	12:45	5.56	4.37	3.570	0.26	94.42
	01/24/97	12:50	4.65	3.63	3.049	0.43	95.34
	01/24/97	12:55	3.47	2.66	2.430	0.42	96.51

This is just a simple example of the power available in the Export command. Many third parties make use of this facility to import MeasureWare data into their tools or data repositories. On MeasureWare for Windows NT there is an optional Graphical user Interface(GUI) for Extract and Utility. This GUI provides for a much easier way to build report layout files.

Correlated Service Management Alarms

The MeasureWare agent is able to send alarm notifications to a number of different event managers:

- PerfView Monitor
- IT/O
- any SNMP Manager
- Execute a local action.

A key point to mention is that the MeasureWare agent is autonomous, that is, there is no polling from a management console. The agent has all the information it needs to make a decision on whether an alarm condition exists, and if so notify the appropriate manager (or execute a local action).

The following is a fragment from an Alarmdef file:

```
alarm GBL_CPU_TOTAL_UTIL > 80 for 10 minutes
type = "test"
start
  red alert "Test Alarm starting"
repeat every 15 minutes
  yellow alert "Test Alarm continuing"
end
  reset alert "Test Alarm ending"
```

As a production alarm the example has limited value, but it does demonstrate a couple of key points. MeasureWare alarms have the concept of DURATION. They are not simple thresholds, in the above example an alarm will not occur until GBL_CPU_TOTAL_UTIL has exceeded 80% for more than 10 minutes. Furthermore the alarm will repeat every fifteen minutes until the condition is cleared.

The following fragment is slightly more complex (and definitely more useful):

```
symptom CPU_Bottleneck type=CPU
rule GBL_CPU_TOTAL_UTIL      > 75    prob 25
rule GBL_CPU_TOTAL_UTIL      > 85    prob 25
rule GBL_CPU_TOTAL_UTIL      > 90    prob 25
rule GBL_RUN_QUEUE           > 2     prob 25

alarm CPU_Bottleneck > 50 for 5 minutes
  type = "CPU"
  start
    if CPU_Bottleneck > 90 then
      red alert "CPU Bottleneck probability= ", CPU_Bottleneck, "%"
    else
      yellow alert "CPU Bottleneck probability= ", CPU_Bottleneck,
"%"
    repeat every 10 minutes
      if CPU_Bottleneck > 90 then
        red alert "CPU Bottleneck probability= ", CPU_Bottleneck, "%"
      else
        yellow alert "CPU Bottleneck probability= ", CPU_Bottleneck,
"%"
    end
    reset alert "End of CPU Bottleneck Alert"
```

The *symptom* construct is able to bring the concept of rules to a MeasureWare alarm. In the above example a variable CPU_Bottleneck has been defined. The value of this variable is set according to the defined rules and then used in a subsequent alarm statement. In the above example if GBL_CPU_TOTAL_UTIL = 87 and GBL_RUN_QUEUE = 3 then CPU_Bottleneck = 75. For each of the rules that is true the *prob* value is accumulated in CPU_Bottleneck.

Alarms may also be constructed using multiple metrics and datasources. For example if we had an Application Response Measurement transaction called FIN_DB_UPDATE_TRANS we could build the following:

```
alarm CPU_Bottleneck > 70 for 5 minutes
  type = "CPU"
  start
    if FIN_DB_UPDATE_TRANS:TT_SLO_PERCENT > 0 then
      red alert "FINANCIAL Update Transaction SLO violation"
    else
      yellow alert "CPU Bottleneck probability= ", CPU_Bottleneck,
"%"
    repeat every 10 minutes
      if FIN_DB_UPDATE_TRANS:TT_SLO_PERCENT > 0 then
        red alert "FINANCIAL Update Transaction SLO violation"
      else
        yellow alert "CPU Bottleneck probability= ", CPU_Bottleneck,
"%"
    end
    reset alert "End of CPU Bottleneck Alert"
```

Here we have built an alarm that will send a red (critical) alert to the event manager if the CPU is busy and the FIN_DB_UPDATE_TRANS is in breach of its Service Level Objective (SLO). If there is no breach of the SLO then the alarm will only send a warning.

It is this ability to correlate a number of different data sources in a functionally rich alarm syntax that makes MeasureWare an ideal choice for building Service Management Alarms. MeasureWare can integrate with any SNMP event manager. However there is turnkey integration between PerfView Monitor, IT/O and MeasureWare. Both PerfView Monitor and IT/O have the ability to launch a context sensitive PerfView Analyzer graph in response to a MeasureWare alarm. There is a similar integration between PerfView Monitor and Network Node Manager.

PerfView

PerfView is the graphical interface into MeasureWare data. It allows for easy analysis of data across the network using a point-and-click interface. PerfView communicates with MeasureWare agents on an as needed basis. There is no network polling and thus no unnecessary network traffic. PerfView incorporates caching algorithms to further reduce the amount of data moved across the network.

One of the key strengths of PerfView is the ability to pull multiple data sources and metrics onto a single screen, or “single pane of glass”. Custom PerfView templates can be constructed for subsequent use in MeasureWare Alarms.

There are three members of the PerfView family:

- PerfView Analyzer
- PerfView Monitor
- PerfView Planner

PerfView Analyzer

The base component of PerfView is the Analyzer. The Analyzer provides a sophisticated interface, allowing for online analysis of data. By simply dragging the mouse across a PerfView graph it is possible to drill down into the detailed data.

PerfView Analyzer has a powerful command line interface. This allows PerfView to be started in a user-defined manner with data sources connected and graphs already displayed. There is also a batch mode, which does not display the user interface. This means that PerfView is very suitable for including in scripts and producing automated reports (See the paper “Using PerfView for Batch Reporting”)⁵. By building the appropriate cgi (Common Gateway Interface) scripts it is possible to publish PerfView data to the Web.

Perfview NT adds a new “what you see is what you get” (*wysiwyg*) print output for all its graphs. In addition there is a new Windows Bitmap (BMP) output mode (along with GIF and PNG). These new output formats for PerfView means that it is even easier to incorporate PerfView graphs into reports, presentations and of course, Web pages.

⁵ Available from <http://www.hp.com/go/rpm>

PerfView Monitor

The PerfView Monitor is one of the options for receiving and managing MeasureWare alarms. By double clicking on any of the alarm messages, a context-sensitive graph will be displayed by PerfView Analyzer. This same level of integration is available without further customization from IT/O and Network Node Manager.

PerfView Monitor can be used alongside IT/O. MeasureWare can send alarms to multiple managers simultaneously. This means that Capacity Planners, who may not want an IT/O console view, can easily review all MeasureWare alarms using a dedicated PerfView monitor interface.

PerfView Planner

PerfView Planner is an easy to use graphical forecasting tool. It makes use of the historical data gathered by the MeasureWare agent(s) across the enterprise. The MeasureWare agent is designed for continuous collection with low overhead. This ensures that it is easy to provide PerfView Planner with the long-term (months rather than days) baseline data required for good statistical forecasting.

Managing a Heterogeneous Environment

Many of the Client/Server environments that we are called upon to manage are mixed platforms. Even if we are lucky enough to have a single vendor shop, and thus only one flavor of UNIX, increasingly we are required to manage a mixed UNIX & Windows NT platforms.

PerfView and MeasureWare provide a consistent interface and user methodology regardless of the platform. There is no need to learn a new tool as you move from one operating system to another. Any differences are optional, not fundamental. For example there is a new GUI on MeasureWare NT, but if you know MeasureWare on UNIX then you can still use all the command line utilities that you are familiar with. MeasureWare provides a consistent metric naming strategy across heterogeneous clients and servers. This helps prevent invalid comparisons between dissimilar metrics. The ability of Perfview to retrieve data from any MeasureWare managed node within the Intranet means easy access to the data needed to manage your environment. There is no need to manage multiple passwords or access methods dependant on platform type.

Summary

The growth of Windows NT and UNIX Client/Server environments bring new management challenges. By providing a consistent view of Performance and Resource Management, PerfView and MeasureWare can put you in control of your enterprise.

Appendix A

The following is a sample MeasureWare parm file for NT

```
# @(#)parm for NT      B.10.02      10/11/96
#
# parm file for performance software
#
# Edit this file as needed to correspond to the specific needs of
# your system.
# See the parm.examples file for additional application suggestions.
# The following control which types of data the scopent process will
# log:
log global process application transaction dev=disk
#
# The following determines thresholds which, when exceeded, will
# cause a process to get logged:
threshold cpu = 10.0
#
# These are some general applications that may apply to your
# system. You should edit them to reflect the applications used on
# your system:
#
# The following processes perform system management functions:
application System
file System smss WINLOGON winlogon lsass spoolss scm ntvdm NTVDM
file EventLog csrss CSRSS progman PROGMAN SERVICES services CONTROL
file nddeagnt netdde clipsrv screg winhlp32 infoview snmp
#
# The following processes are common programs run on desktop
# workstations:
application DeskTop
file 123* flw* WINWORD* excel* msaccess*
file CMD NTBACKUP FTP XCOPY NOTEPAD TERMINAL terminal WINFILE CLOCK
file rx RX filemgr PERFMON
#
# This application may be appropriate if you want to monitor the
# overhead associated with the performance tools:
application PerfTools
file midaemon* ttd* scope* rpcd
file alarm* agdb* perf* rep_server extract utility mwa mwrptrw
file dsilog sdl* pv*
```