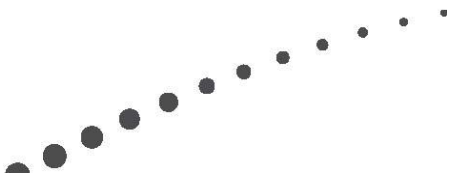
XenServer Performance Monitoring for Scalability Testing

This informational whitepaper provides an overview of performance monitoring tools and techniques for XenServer and a description of scripting methods to gather customized performance metrics.



CITRIX®

Contents

Overview

Citrix XenServer is used widely as a high-performance server and desktop hardware virtualization solution for both large and small organizations. Citrix XenServer's management application, XenCenter, is used to monitor the performance of an environment in real-time and historically. For performance and scalability testing purposes, a more-customized log of XenServer performance may be necessary. Citrix Worldwide Consulting Solutions conducts in-house, cross-product scalability and performance testing to provide guidance to customers. This document is a result of such efforts and provides an overview of the different ways of monitoring XenServer resource utilization for testing purposes, as well as a sample template for creating customized scripts.

Resource Monitoring on XenServer

The two most common means of monitoring the performance of XenServer are using XenCenter and running scripts from the XenServer command line interface (CLI). An overview of these two methods is given in this section.

Using XenCenter to Collect Performance Data

Citrix XenServer provides the ability to monitor important system resources of the hosting infrastructure servers and virtual machines in real-time from XenCenter, which is a graphical user interface (GUI) used for administrative purposes. Performance data about each XenServer is automatically collected about itself by default and displayed within XenCenter. XenServer also collects performance data for each virtual machine that it is hosting. Statistics are gathered and stored at different granularities: 5 seconds for the past 10 minutes, 1 minute for the past 2 hours, 1 hour for the past week and 1 day for the past year.

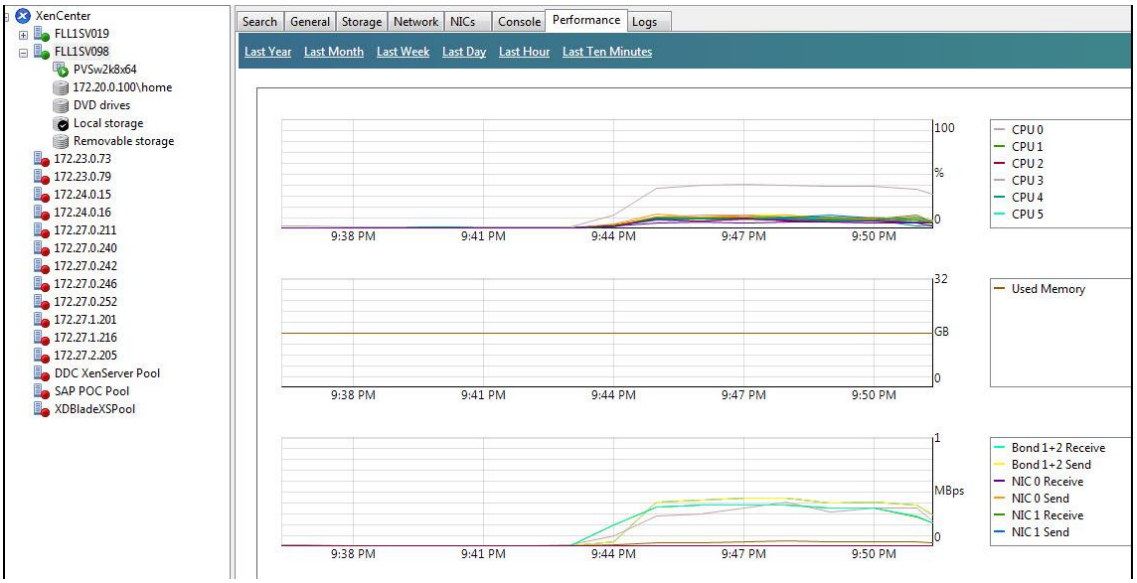


Figure 1: XenServer Performance Information as viewed in XenCenter

This performance data is also used as part of the Dynamic Workload Balancing feature of Essentials for XenServer, which ensures optimal utilization of physical resource pools for balancing virtual workloads. Upgrading to Essentials for XenServer allows administrators to best plan and enhance a XenServer environment, simply by viewing intelligent host recommendations within XenCenter.

The information displayed by XenCenter offers an informative glance at the performance of a XenServer for administrative purposes, but may be unwieldy for use cases such as refined performance testing or customized historical trending. The data cannot be readily exported, the granularity cannot be easily changed and some types of information are not gathered. The remaining sections of this document describe alternative means of achieving these goals.

Using the XenServer CLI for Performance Data Capture

The XenServer command-line interface (CLI) offers the ability to query the hypervisor for performance information about itself and to use standard Linux tools and utilities to draw performance data, as well. There are many more types of performance data that can be acquired from the XenServer CLI than from XenCenter. This is especially useful during performance and scalability testing for capturing customized sources of data and recording fine-grained results for later analysis. This document describes the various ways that this can be accomplished, including creating a customized performance monitoring script and using both built-in and third-party tools and utilities.

Creating a Performance Monitoring Script

This section describes the pre-requisite knowledge needed in order to create a monitoring script for XenServer in order to capture customized performance data. Scripting examples and XenServer CLI outputs are shown for a better understanding of these topics. A sample script template is also provided that can be tailored for customized XenServer performance data collection.

Overview of Bash Scripting

The XenServer command line interface (CLI) is a Linux Bash shell that interprets both Linux and XenServer-specific commands, and also allows for the creation of variables, loops and basic programming functions. A Bash script is simply a text file that the XenServer's Bash shell interprets line-by-line. XenServer commands that display performance statistics can be integrated into a simple Bash script to provide a customized, lightweight monitoring capability for any XenServer.

The remainder of this section describes the pre-requisite knowledge and creation of a basic Bash script that monitors a XenServer's CPU, memory and network interface card (NIC) resources.

Familiarity with using a Linux/UNIX prompt and the built-in "vi" text editor, or other text editor, is recommended.

Creating and Executing a Bash Script

As previously mentioned, a Bash script is a plain text file that the XenServer CLI interprets line-by-line for execution, as if text commands were entered directly into the Linux shell. A Bash script should be saved with a ".sh" file extension. Bash scripts can be run on the XenServer CLI by typing "sh" followed by the filename. Should a Bash script fail to complete on its own, it can be stopped with "Ctrl-C" on the keyboard. The following example demonstrates the execution of a Bash script from the XenServer CLI.

```
Input Command
[root@fll1sv010 ~]# sh utilisation.sh
```

```
Resulting Output

"Time","Processor 0","Processor 1","Processor 2","Processor3","Memory Bytes
Used","Eth0 Bytes Sent","Eth0 Bytes Received"

"17:23:00","0.045","0.000","0.000","0.000","1318703104","273.452301","367.7
10999"
"17:23:10","0.066","0.000","0.000","0.000","1318703104","980.801392","239.9
36356"
"17:23:20","0.004","0.000","0.000","0.000","1318703104","115.527084","74.41
4482"
    <metrics continue until Ctrl-C is pressed>
```

Figure 2: Running the Performance Monitoring Bash Script

Selecting Applicable Performance Metrics

The most common types of performance metrics can be gathered from XenServer in a straight-forward manner. XenServer automatically collects a variety of performance data on both physical hosts and virtual machines. Data on resources such as CPU, memory and network interface cards (NIC) are readily available for both physical hosts and virtual machines and can be gathered by creating queries on the XenServer CLI.

The "data-source" methods can be performed on both hosts and virtual machines for gathering such data. XenServer documentation, as shown in the References section, describes all of the available commands that can be used for such purposes, but the example in Figure 3 demonstrates how to query the XenServer for a list of available metrics for hosts and virtual machines.

In the example below, the "xe host-data-source-list" command is used to display the available performance statistics that can be retrieved for the XenServer host, which is named fllsv010. Information such as CPU, memory, NIC data and more are displayed as a result of using this command so that they can then be incorporated into a customized metrics script.

```
Input Command
[root@fll1sv010 ~]# xe host-data-source-list
```

```
Resulting Output
          name_label: vbd_hda_write_latency
    name_description: Reads from device 'hda' in microseconds
             enabled: false
            standard: false
                 min: 0.000
                 max: nan
               units: microseconds

          name_label: cpu0
    name_description: Physical cpu usage for cpu 0
             enabled: true
            standard: true
                 min: 0.000
                 max: 1.000
    <metrics continue…>
```

Figure 3: XenServer CLI Querying of Performance Data

The example above is specific to a XenServer host, but the same syntax can also be used to gather the available performance statistics for a virtual machine. The "xe vm-data-source-list" command can be performed on a particular virtual machine to find the list of available metrics. The outputs of these and other queries can be included in a script to poll the XenServer or virtual machine for data at specific intervals. This process is described in the remainder of this document.

Creating Variables and Loops for Continuous Metric Polling

In order to capture and store data from XenServer, variables and programming loops must be used within a script. The script excerpt contained in the next figure is provided for demonstrating these concepts, followed by an explanation.

```
5 count=0
6 #set up a never-ending loop
7 while [ $count -lt 2 ]
8 do
9 #refresh all data
...
<script continues>
...
memUsed'","'$pifEth0Sent'","'$pifEth0Received'"' >> \perflogXS.csv
33
34         sleep 10
35#count=$(($count + 1))
36 done
```

Figure 4: Do-While Loop Example

The loop used in the example above is a "do-while" loop, which begins on line 7 and ends on line 36. The "count" variable is used to keep track of the number of iterations the loop has been through and is first created and set to zero on line 5. The conditions for either continuing or ending this loop are set on line 7. As seen on line 7, as long as the variable "count" is less than the number two, the loop will run. These conditions could be altered so that the loop runs a finite number of times, before ceasing, by manipulating the starting variable "count" on line 5, the number two on line 7 and the commented section on line 35, which increments the count variable each iteration through the loop.

Lines 34 through 36 complete this loop. Line 36 marks the end of the loop with the "done" statement. The "#" character at the start of a line is a comment and indicates that the line should not be executed on the XenServer command line. The "#" can be removed on line 35 in order to increment the count variable so as to end the running of this script automatically. Line 34 uses the "sleep" command to pause the loop for ten seconds. The number "10" can be changed for any number of seconds in order to set the desired polling interval.

Using these basic variables and loops can be useful in scripting for accomplishing the tasks necessary to record and manipulate XenServer performance data, as shown in the successive sections of this document.

Retrieving the Output of XenServer CLI Commands

Some XenServer CLI commands require inputs that are unique to each individual XenServer host or virtual machine. These properties must first be gathered, before being entered into such a script. In the example script given in this section, the storing of command line outputs is demonstrated in this context.

In the script excerpt below, the universal unique identifiers (UUID) of the physical CPU cores are required as inputs to several XenServer commands. The UUIDs are first gathered using the "xe

host-cpu-list" command for use in XenServer CLI resource utilization query commands. On line 14 of this script, a sample of the CPU utilization of the physical core numbered zero is calculated using the "xe host-cpu-param-get" command, as shown below. The output of this sample is stored in the variable "procUtil0" and is later retrieved for outputting to the CSV file on line 32. The formatting of each of these operations should be noted, including the parentheses, single and double quotation marks and "$" characters. This method of storing and retrieving resource utilization samples is used extensively in such scripting.

```
13          #calculates processor utilisation, uuid's hard-coded
14          procUtil0="$(xe host-cpu-param-get uuid=96af1333-fe7a-fdd8-28f1-
e04e43e96319 param-name=utilisation)"
15          procUtil1="$(xe host-cpu-param-get uuid=d90731d8-383f-3357-97b0-
3865e07f7207 param-name=utilisation)"
16          procUtil2="$(xe host-cpu-param-get uuid=a50dc576-76f4-68ee-5bb2-
989c784263bd param-name=utilisation)"
17          procUtil3="$(xe host-cpu-param-get uuid=9c1d1068-e68c-2e13-6313-
6236ba421423 param-name=utilisation)"
18
...
<script continues>
...
32          echo
'"'$dateVar'","'$procUtil0'","'$procUtil1'","'$procUtil2'","'$procUtil3'","
'$memUsed'","'$pifEth0Sent'","'$pifEth0Received'"' >> \perflogXS.csv
```

Figure 5: Using Variables and Retrieving CLI Data

Creating an Output File of the Metrics

Performance metrics that are outputted to the XenServer console, as shown in the previous example, can be stored to a log file for later use. The command-line entry shown below demonstrates this functionality. In this example, "xe host-data-source-list > out.txt", stores the output of the data-source query command to a text file that can be read using a text editor for easy viewing. The built-in Linux utility, "vi", is then used to view the file.

```
Input Command
[root@fll1sv010 ~]# xe host-data-source-list > out.txt
```

```
Input Command
[root@fll1sv010 ~]# vi out.txt
```

Figure 6: XenServer CLI commands for File Output and Viewing

Example Scripts

The information used in the previous sections can be used to create a basic monitoring script for XenServer, as shown in the example in this section. This example script is provided as a template for expansion to suit the needs of performance and scalability testing. The script outputs the CPU utilization of four physical cores, the memory usage, and the physical network adapter usage. This information is formatted and stored as a CSV file, which can be used for analysis and graphing. Scripts such as these are especially useful for creating customized graphs that combine information about both hosts and virtual machines side-by-side.

The sample script is shown below in its entirety with line numbers designated on the left side. Comments and lines that are not executed on the command line begin with a "#" character or are simply left blank. Variables are created by using an "=" character and are retrieved by using the "$" character. The "echo" command outputs information to the screen, and the ">" and ">>" character entries send output to a file (overwrite and append, respectively). Each line is executed in succession on the XenServer CLI as if it were inputted manually by a user, as described previously.

```
1 #define the column titles of the CSV file here, overwrite if need be
2 echo '"Time","Processor Zero","Processor One","Processor Two","Processor
Three","Memory Bytes Used","Eth0 Bytes Sent","Eth0 Bytes Received"' >
perflogXS.csv
3 echo '"Time","Processor Zero","Processor One","Processor Two","Processor
Three","Memory Bytes Used","Eth0 Bytes Sent","Eth0 Bytes Received"'
4 echo ""
5 count=0
6 #set up a never-ending loop
7 while [ $count -lt 2 ]
8 do
9 #refresh all data
10         #calculates date & time for the current sample
11         dateVar="$(date +'%H:%M:%S')"
12
13         #calculates processor utilisation, uuid's hard-coded
14         procUtil0="$(xe host-cpu-param-get uuid=96af1333-fe7a-fdd8-28f1-
e04e43e96319 param-name=utilisation)"
15         procUtil1="$(xe host-cpu-param-get uuid=d90731d8-383f-3357-97b0-
3865e07f7207 param-name=utilisation)"
16         procUtil2="$(xe host-cpu-param-get uuid=a50dc576-76f4-68ee-5bb2-
989c784263bd param-name=utilisation)"
17         procUtil3="$(xe host-cpu-param-get uuid=9c1d1068-e68c-2e13-6313-
6236ba421423 param-name=utilisation)"
18
19         #calculates memory utilisation by subtracting bytes free from
bytes total
20         memFree="$(xe host-list params=memory-free --minimal)"
21         memTotal="$(xe host-list params=memory-total --minimal)"
22         memUsed=$(($memTotal - $memFree))
23
24         #retrieves NIC bytes sent/received
25         pifEth0Sent="$(xe host-data-source-query data-
source=pif_eth0_tx)"
26         pifEth0Received="$(xe host-data-source-query data-
source=pif_eth0_rx)"
27
28 #now output to screen
29         echo
'"'$dateVar'","'$procUtil0'","'$procUtil1'","'$procUtil2'","'$procUtil3'","
'$memUsed'","'$pifEth0Sent'","'$pifEth0Received'"'
30
31 #now append to the file
32         echo
'"'$dateVar'","'$procUtil0'","'$procUtil1'","'$procUtil2'","'$procUtil3'","
'$memUsed'","'$pifEth0Sent'","'$pifEth0Received'"' >> \perflogXS.csv
33
34         sleep 10
35 #count=$(($count + 1))
36 done
```

Figure 7: Basic XenServer Resource Monitoring Bash Script

Other Types of Scripting

Performance monitoring on XenServer is not limited to scripting on the XenServer CLI, but can also be attained by using more advanced methods. Many of the built-in Linux tools are written and compiled in a variety of programming languages. Creating more robust monitoring programs typically requires computer programming knowledge, such as that in Python, Perl, C or others. Using these methods require much more time and effort, but may provide greater functionality in circumstances where:

- Tighter shell integration or better performance is more beneficial
- Enhanced text editing or I/O formatting is needed
- Monitoring multiple XenServers or virtual machines requires more efficient coding practices

Many such programming languages can be interpreted and compiled on XenServer.

Controlling and Monitoring a XenServer Environment for Performance Testing

Executing a performance monitoring script on a single XenServer and retrieving the results is straight-forward, but conducting such a task across multiple XenServers presents its challenges. This section describes additional considerations for such a scenario and tools that can be of assistance.

Environment Considerations

The performance monitoring script template shown in this document does not consume noticeable physical resources to run on a XenServer. However, additional computing overhead may occur if the polling frequency is increased or if additional machine data is collected. Observations should be taken to ensure that such monitoring scripts do not impacting the systems under test.

Another consideration for a test environment is to ensure that all of the XenServers have synced system clocks so that data from the different sources can be overlaid properly during interpretation and analysis. XenServer supports Network Time Protocol (NTP) syncing, which can be set up from the XenServer CLI or console. Similarly-formatted data files and formats are also prudent in this regard.

Retrieving and Moving Log Files

By default, XenServer can be accessed with a command-line interpreter and a file transfer client. To upload and download scripts and log files from a XenServer, two freely-available utilities are most often used: WinSCP and FileZilla. These two clients utilize the Secure Shell protocol to log into a XenServer and use the SCP and SFTP file transfer protocols to upload and download files. Links to download these utilities are available in the References section of this document.

To move files directly between different XenServers in an environment, there is a built-in Linux utility on XenServer called "secure copy", which is a remote file copy program. This program uses SSH to log into another XenServer and upload or download files. More information about this utility and instructions on how to use it can be found by typing "man scp" on the XenServer CLI. An example of using SCP to upload a file to another XenServer can be seen in the example below:

Input Command

```
[root@fll1sv010 ~]# scp perflogXS.csv root@172.27.2.205:/root/
```

Resulting Output

```
perflogXS.csv                                 100%  289     0.3KB/s   00:00
```

Figure 8: Using the Linux SCP Utility for File Transfer between XenServers

Executing Scripts on Multiple XenServers

There are several methods of executing commands and running scripts on many XenServers in an environment. One simple method of enabling script execution simultaneously on different XenServers is to put the XenServers in the same pool and then modify the script to include the names of the hosts. Another easy method is to use the "cron" Linux utility on each XenServer, which can schedule execution of the script at a specific time. A third method of executing scripts on multiple XenServer simultaneously is to script SSH logins to each XenServer within a script. Using a password-free login can be set up for this so as to streamline remote script execution.

For increased control and flexibility on any number of XenServers, third-party tools can be used. One such tool that has been used successfully by Citrix Consulting to log into multiple XenServers simultaneously and initiate scripts is called Capistrano, which can be downloaded freely. A link is contained in the References section of this document and an example of the use of Capistrano is shown below.

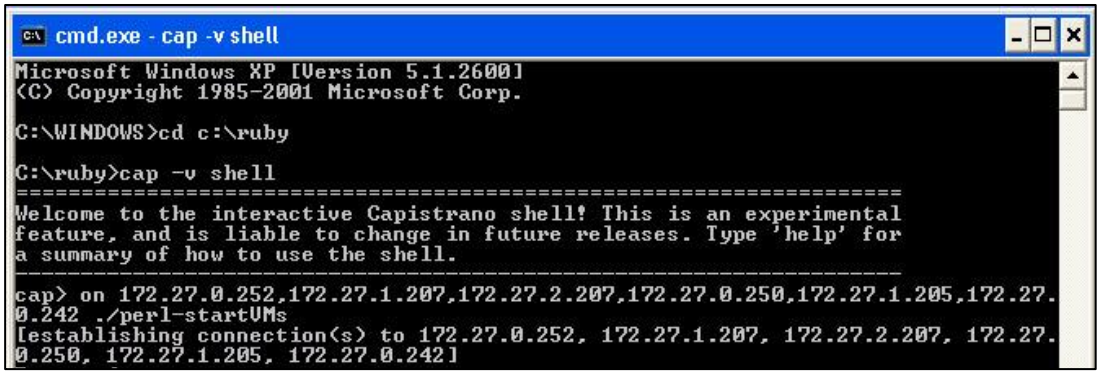


Figure 9: Using a Third-Party Utility for Simultaneous Script Execution across XenServers

In the example above, the Capistrano shell is being utilized to access multiple XenServers from a centralized CLI. The "on" command is followed by the IP address of each XenServer where the command will be executed. Following all the XenServer IP addresses is the actual command that will be executed on each server simultaneously. It should be mentioned that if the administrator is executing a script on all the servers utilizing the Capistrano shell, the script needs to be located in the same folder structure for every server.

Additional Performance Monitoring Utilities

Built-in Linux Utilities

XenServer is built on the Linux operating system, which comes with pre-installed, command-line resource monitoring utilities that can be exceptionally valuable. The following list provides several examples of useful utilities for performance testing that are built-into the XenServer Linux distribution:

- mpstat – reports processor-related statistics
- vmstat – reports virtual memory statistics
- netstat – reports network-related statistics
- iostat – reports input/output statistics for devices and partitions

The types of data that can be drawn from these utilities and others can be customized to suit the needs of performance testing, should more precision be needed. Listed below are a few useful examples of Linux CLI entries that may be of help during performance and scalability testing on a XenServer.

- Find read/write statistics to/from NFS directory: iostat -n
- Output detailed packet information for TCP and UDP traffic: netstat -s

- View the XenServer processor queue length: sar -q 1 0

These utilities can be individually used to poll the XenServer for such statistics and output to the screen in real-time. The following examples show the iostat tool used to output disk I/O statistics and the mpstat tool used to output CPU statistics.

Input Command

```
[root@fll1sv010 ~]# iostat 5
```

Resulting Output

```
Linux 2.6.18-128.1.6.el5.xs5.5.0.496.1012xen (fll1sv010)        10/12/2009

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           0.42    0.00    0.77    0.01    0.03   98.77

Device:            tps   Blk_read/s   Blk_wrtn/s   Blk_read   Blk_wrtn
cciss/c0d0        2.60         4.35        40.87    7956356   74748880
cciss/c0d0p1      1.55         0.00         0.00          0          0
cciss/c0d0p2      0.01         0.00         0.00          0          0
cciss/c0d0p3      1.04         0.00         0.00          0          0
dm-0              0.01         0.02         0.16      41074     299195

<output continues>
```

Figure 10: Resource Monitoring with the Built-in Linux Tool iostat

Input Command

```
[root@fll1sv010 ~]# mpstat 5
```

Resulting Output

```
Linux 2.6.18-128.1.6.el5.xs5.5.0.496.1012xen (fll1sv010)        10/12/2009

03:11:23 PM  CPU   %user   %nice    %sys %iowait    %irq   %soft  %steal
%idle    intr/s
03:11:28 PM  all   31.69    0.00   63.19    0.00    0.00    0.98    0.00
4.13   3066.14
03:11:33 PM  all   26.15    0.00   65.07    0.00    0.00    0.40    0.20
8.18   2042.71
03:11:38 PM  all    8.96    0.00   24.30    0.00    0.00    0.00    1.39
65.34    599.60
03:11:43 PM  all    0.40    0.00    0.40    0.00    0.00    0.00    0.60
98.60    126.95
03:11:48 PM  all    0.60    0.00    0.80    0.00    0.00    0.00    0.80
97.80    403.99
03:11:53 PM  all    7.39    0.00    4.39    0.00    0.00    0.00    0.60
87.62   3649.90
<output continues>
```

Figure 11: Resource Monitoring with the Built-in Linux Tool mpstat

Third-Party Resource Monitoring Utilities

There are a variety of freely-available tools that can be downloaded and installed on XenServer. Although doing so may modify the XenServer in a manner that could make it unsupported, these tools may offer increased flexibility for testing purposes by:

- Providing resource information that was previously irretrievable by the XenServer CLI commands or by built-in Linux utilities.

- Returning statistics in a more-easily digestible format for data analysis, such as in the form of a spreadsheet or a particular database-format file, without the need for Linux scripting know-how or script customization.

One such utility that has been successfully used for recording XenServer performance statistics is called "dstat", which can be downloaded freely. The following two commands can be used in succession to download and install the dstat tool from a third-party, non-Citrix-associated, online repository and install it automatically on XenServer:

1. rpm -Uhv http://apt.sw.be/redhat/el5/en/i386/rpmforge/RPMS/rpmforge-release-0.3.6-1.el5.rf.i386.rpm

2. yum install dstat

The installation of "dstat" can be found in the following example, which shows a screen capture of the XenServer CLI while performing these tasks.

```
[root@FLL1SV011 ~]# rpm -Uhv
http://apt.sw.be/redhat/el5/en/i386/rpmforge/RPMS/rpmforge-release-0.3.6-
1.el5.rf.i386.rpm
warning: /var/tmp/rpm-xfer.3tdOIq: Header V3 DSA signature: NOKEY, key ID
6b8d79e6
Preparing...                ########################################### [100%]
   1:rpmforge-release       ########################################### [100%]
[root@FLL1SV011 ~]# yum install dstat
Loading "fastestmirror" plugin
Determining fastest mirrors
 * rpmforge: apt.sw.be
 * citrix: updates.vmd.citrix.com
rpmforge                  100% |=========================| 1.1 kB    00:00
primary.xml.gz            100% |=========================| 3.5 MB    00:55
rpmforge  : ################################################## 9615/9615
citrix                    100% |=========================|  951 B    00:00
primary.xml.gz            100% |=========================|  237 B    00:00
Setting up Install Process
Parsing package install arguments
Resolving Dependencies
--> Running transaction check
---> Package dstat.noarch 0:0.6.9-1.el5.rf set to be updated
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================
 Package                 Arch       Version          Repository        Size
================================================================================
Installing:
 dstat                   noarch     0.6.9-1.el5.rf   rpmforge          192 k

Transaction Summary
================================================================================
Install      1 Package(s)
Update       0 Package(s)
Remove       0 Package(s)

Total download size: 192 k
Is this ok [y/N]: y
Downloading Packages:
(1/1): dstat-0.6.9-1.el5. 100% |=========================| 192 kB    00:03
Running rpm_check_debug
Running Transaction Test
Finished Transaction Test
Transaction Test Succeeded
Running Transaction
  Installing: dstat                         ######################### [1/1]

Installed: dstat.noarch 0:0.6.9-1.el5.rf
Complete!
```

Figure 12: Third-Party Resource Monitoring Tool Installation

The next example demonstrates the use of dstat, which aggregates and records data to CSV file format from various, built-in Linux utilities, such as iostat and mpstat. In this example, CPU, physical disk, network interface card and memory statistics are polled every second and outputted to both the screen and to the CSV file. Third-party tools such as these are especially useful for being able to quickly set up a robust, lightweight monitoring capability, albeit potentially in an unsupported fashion.

Input Command

```
[root@FLL1SV011 ~]# dstat --output PerformanceOutput.csv
```

Resulting Output

```
----total-cpu-usage---- -dsk/total- -net/total- ---paging-- ---system--
usr sys idl wai hiq siq| read  writ| recv  send|  in   out | int   csw
  0   1  99   0   0   0|1448B   18k|   0     0 |   0   0.1 |  71   996
  0   0 100   0   0   0|   0     0 |1034B 2570B|   0     0 |  51    73
  0   0 100   0   0   0|   0     0 |1437B  998B|   0     0 |  39    57
<output continues>
```

Figure 13: Resource Monitoring using the Third-Party Tool dstat

Summary

Analyzing the performance of XenServer for scalability and testing purposes is best achieved through the use of Bash scripts, built-in Linux utilities and third-party tools. The information contained in this document provides the reader with a foundation of knowledge in accomplishing this goal.

CITRIX®

Appendix A: Sample XenServer Performance Monitoring Bash Script

The following Bash script has been used successfully by Citrix Consulting to monitor the CPU, memory and NIC utilization of a XenServer host. Line numbers are provided for easier viewing.

```
1 # 2008 - 2010 Citrix Consulting
2 # This is a sample script that collects a XenServer's resource utilization
3 # and outputs it to a .csv file, which can be viewed and analyzed as a
spreadsheet
4 # In this example, there are four CPU cores on the hardware, which means
5 # that four CPU UUID's must be found in order to gather CPU utilization for each
6 # This can be done by typing "xe host-cpu-list" on the command-line interface
(CLI)
7 # The same must be done for the network adapters by typing "xe pif-list" on the
CLI
8 # The output file will appear in the same directory, and is overwritten each time
9 # This script collects CPU, memory, and the number of running VMs every few
seconds
10 # This script also collects network data, including bytes sent/received on NICs
11 # To change the sample rate, change "sleep" variable at the bottom of the script
12
13 #define the column titles of the CSV file here
14 echo '"Time","Processor 0","Processor 1","Processor 2","Processor3","Memory
Bytes Used","Eth0 Bytes Sent","Eth0 Bytes Received"' > perflogXS.csv
15 echo '"Time","Processor 0","Processor 1","Processor 2","Processor3","Memory
Bytes Used","Eth0 Bytes Sent","Eth0 Bytes Received"'
16 echo ""
17 count=0
18 #set up a never-ending loop
19 while [ $count -lt 2 ]
20 do
21 #refresh all data
22     #calculates date & time for the current sample
23     dateVar="$(date +'%H:%M:%S')"
24
25     #calculates processor utilisation, uuid's hard-coded for now...
26     procUtil0="$(xe host-cpu-param-get uuid=96af1333-fe7a-fdd8-28f1-e04e43e96319
param-name=utilisation)"
27     procUtil1="$(xe host-cpu-param-get uuid=d90731d8-383f-3357-97b0-3865e07f7207
param-name=utilisation)"
28     procUtil2="$(xe host-cpu-param-get uuid=a50dc576-76f4-68ee-5bb2-989c784263bd
param-name=utilisation)"
29     procUtil3="$(xe host-cpu-param-get uuid=9c1d1068-e68c-2e13-6313-6236ba421423
param-name=utilisation)"
30
31     #calculates memory utilisation by subtracting bytes free from bytes total
32     memFree="$(xe host-list params=memory-free --minimal)"
33     memTotal="$(xe host-list params=memory-total --minimal)"
34     memUsed=$(($memTotal - $memFree))
35
```

Figure 14: Sample Performance Monitoring Script (Figure 1 of 2)

```
36     #retrieves NIC bytes sent/received
37     pifEth0Sent="$(xe host-data-source-query data-source=pif_eth0_tx)"
38     pifEth0Received="$(xe host-data-source-query data-source=pif_eth0_rx)"
39
40 #now output to screen
41     echo
'"'$dateVar'","'$procUtil0'","'$procUtil1'","'$procUtil2'","'$procUtil3'","'$memUse
d'","'$pifEth0Sent'","'$pifEth0Received'"'
42
43 #now output to the file
44     echo
'"'$dateVar'","'$procUtil0'","'$procUtil1'","'$procUtil2'","'$procUtil3'","'$memUse
d'","'$pifEth0Sent'","'$pifEth0Received'"' >> \perflogXS.csv
45
46     sleep 10
47 #count=$(($count + 1))
48 done
```

Figure 15: Sample Performance Monitoring Script (Figure 2 of 2)

References

Capistrano: http://www.capify.org/index.php/Capistrano

File transfer clients:

- FileZilla http://filezilla-project.org/
- WinSCP http://sourceforge.net/projects/winscp/

Third-party Linux system monitoring utility dstat: http://dag.wieers.com/home-made/dstat/

XenServer Software Development Kit Guide: http://support.citrix.com/article/ctx118450

XenServer 5.5 Administrators Guide: http://support.citrix.com/article/CTX121745

Revision	Change Description	Updated By	Date
0.1	First draft	Zachary Menegakis	October 8, 2009
0.2	Quality Assurance Review	Carisa Stringer	October 13, 2009
0.3	Revision	Consulting Solutions	January 13, 2010
1.0	Final draft	Zachary Menegakis	February 5, 2010

About Citrix

Citrix Systems, Inc. (NASDAQ:CTXS) is the leading provider of virtualization, networking and software as a service technologies for more than 230,000 organizations worldwide. It's Citrix Delivery Center, Citrix Cloud Center (C3) and Citrix Online Services product families radically simplify computing for millions of users, delivering applications as an on-demand service to any user, in any location on any device. Citrix customers include the world's largest Internet companies, 99 percent of Fortune Global 500 enterprises, and hundreds of thousands of small businesses and prosumers worldwide. Citrix partners with over 10,000 companies worldwide in more than 100 countries. Founded in 1989, annual revenue in 2008 was $1.6 billion.