**CSI3670**

**Winter 2021**

**Lab 7**

**Due date: April 18th @ 11:59pm**

**Synopsis**

One of the major issues with servers is ensuring that performance is up to snuff. In this lab you will (1) create a PowerShell script to use a lot of resources and (2) use common performance analyzation/tuning tools to monitor it. Also, lots of clicking.

**Walkthrough**

***Don’t forget, if you don’t know how to use a command or want more information using cmd.exe, you can typically run <command> /? to get the list of arguments. (PowerShell is Get-Help <command>)***

First, let’s create a PowerShell script.

SLOW.ps1:

$total = 1000;

for ($i = 0; $i -le $total; $i++)

{

Write-Progress -Activity Testing -Status “Just slowing things down” -PercentComplete (($i / $total) \* 100);

}

If you do a direct copy/paste, you may need to manually enter the double quotes to avoid errors. Save it and run it with $total set to 1000 (as shown). Make sure that the progress bar is being displayed appropriately as you run it. It might be too fast for you to see, but it should be displayed in your PowerShell terminal

Now, bump $total up to 100000. **Take a screenshot of it working and paste into Q5. If your VM completes it too quickly to take a screenshot, add another 0 to total.**

Open up a command prompt (cmd.exe, not PowerShell). Run the following commands (without the (1)).

(1) tasklist /fi "memusage ge 15000"

(2) tasklist /fi "cputime ge 00:01:00"

**Take a screenshot of each output and paste it into Q5. Describe what each command does in Q1.**

We can also use the command line to autostart services using the sc command (or start, restart, stop, etc.). For instance, we can have the Windows Remote Management service auto-start whenever you login.

sc config winrm start= demand

We can also use cmd to kill programs. Run your SLOW.ps1 script in PowerShell and then run the following command in Command Prompt (if this command doesn’t work, replace pwsh.exe with powershell.exe or powershell\_ise.exe):

taskkill /im pwsh.exe

Let’s assume your machine is acting up. Time to reboot. Issue the following command from the command prompt:

shutdown /r /t 90 /f

Open up another command prompt when Windows comes back. Let’s gather some server information. Run the following command in a Command Prompt terminal:

systeminfo /fo list

**Take a screenshot of the top page of input (its going to be a large list...you don’t need to show the whole thing) and paste into Q5.**

Our last command prompt command that we’ll play with is typeperf. This gives us access to the various performance counters that we can see. The following command will actually give us the names of all the counters available to us:

typeperf -q > counters.txt

**Open up counters.txt with a text editor. Pick 3 of the counters from different sections (eg., don’t describe three of the counters specific to clusters only) and describe what they are (look up what they do) in Q2.**

First let’s watch processor usage:

typeperf “\Processor Information(\*)\% Processor Time”

This will run continuously until you hit Ctrl+C. **Take a screenshot and paste into Q5.**

**What does the following command do differently than the prior command? Answer in Q3.**

typeperf “\Processor Information(\*)\% Processor Time” -sc 10 -si 5

We can also check how much available memory exists. Run the following:

typeperf “\Memory\Available Mbytes” -sc 10 -si 5

Another way to monitor performance is with the **Resource Monitor**. This is pretty similar to the task manager, but monitors resources (naturally). You can open it from Server Manager → Tools → Resource Monitor (or by searching for Resource Monitor in the Start Menu or running resmon from the Run prompt).

Graphical user interface, application

Description automatically generated

Other than the typical panels showing us disk usage, memory usage, etc., we can troubleshoot programs that are not responding. Right-click on any process and then click Analyze Wait Chain. This will tell you what the process is waiting for.

**Take a screenshot of a random service that you checked and paste into Q5.**

Lastly, let’s look at the Performance Monitor. This is basically a nice GUI that lives on top of the System Monitor. In Server Manager, under Tools, click Performance Monitor.

Open it up and expand Performance, Monitoring Tools on the left (as shown). Note that you’re watching the live performance of your processor (by default).

Graphical user interface, application, Word

Description automatically generated

Click the little plus icon :

The performance counters we can watch are the same as before.

Expand Memory, then Add Available Bytes and Pages/sec.

Graphical user interface, application

Description automatically generated

Click OK. Right click on each of the newly-added counters and change their color. Take a screenshot of your performance graph with **three distinct colors for your three distinct performance counters and paste into Q5.**

Now, run your PowerShell script again, give it a moment, and then **take another screenshot to demonstrate how your script impacts your system, pasting into Q5.**

Either cancel your script or wait for it to end.

So far we’ve watched live performance data, but what if we want to log it (for instance, to define a baseline of *normal* system activity). Expand **Data Collector Sets** in your Performance Monitor window**.** Right-click on **User Defined** and add a new **Data Collector Set.**

**Graphical user interface, text, application

Description automatically generated**

Give it a name and leave **Create from a template selected.** Select **System Performance**. Accept the default location and Run as: option. Click Finish. Double click your new data collector set (DCS) and double click on Performance Counter. Change the Log format to Comma Separated:

Graphical user interface, application

Description automatically generated

Click the File tab and you can see where the file will be saved. You can also change the file name if so desired in the top text box. Click Apply and OK to close. Right-click on your created DCS and click Properties. Click the Stop Condition tab. Change Overall duration to 5 minutes. Click Apply and OK.

Graphical user interface, text, application

Description automatically generated

Right-click on your DCS again and click Start. Wait for 5 minutes and then view your report. You do so by expanding Reports → User Defined → <YOUR DCS> on the left. Right click and View Report when it has stopped running (you can check by looking at the little play icon on your DCS).

Graphical user interface, text, application

Description automatically generated

**Take a screenshot of your report and paste into Q5.**

**Lab Report (90 points)**

1. (10 points) What do the tasklist commands do as asked on the first page?
2. (15 points) Describe the counters as asked.
3. (5 points) Answer the question about typeperf:
4. (15 points) Only one real question outside of what was previously asked. What is a performance baseline and why is it necessary, and furthermore, how would you go about creating one (i.e., what you just did in the lab)? Make sure to address all three points.
5. (45 points, 5 points each) Paste all screenshots here. There should be at least 8, or 9 if you did two separate screenshots at the end of page 1.