International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 03 Issue: 03 | Mar-2016 www.irjet.net p-ISSN: 2395-0072

Performance Evaluation of any Application with C# and R

Devharsh Trivedi

M.Tech. student, Computer Science and Engineering (Information and Network Security), Institute of Technology, Nirma University, Ahmedabad, India.

Abstract – *Aim of this paper is to present a framework that helps visualize any program's performance on a high end multi core system. To achieve this purpose two things are required: (i) A mechanism to log the data and (ii) A mechanism to plot the data. I have presented two individual applications for logging system statistics in C# with windows forms and R shiny application for plotting.*

Key Words: Profiler, Shiny, R, C#

1. INTRODUCTION

When it comes to profiling an application we focus on two core things i.e. CPU and Memory usage and we try to utilize spare amount as much as we can. My aim was to create a simple tool that can help visualize application performance on various multi-cores machines easily. I have used existing open source and freeware tools and technologies to build my own mechanism.

2. TOOLS AND TECHNOLOGIES

IDE – SharpDevelop, RStudio

Languages – C#, R, Javascript/HTML

Libraries/References – System.Diagnostics, System.IO, System.Management, Shiny, Shinydashboard, Shinyjs, Plotly

2.1. LOGGING

I have used windows forms application for GUI and logic is in C#. I have used Microsoft's references like Diagnostics and Management that provides CPU and Memory usage information. Parameters I am logging for Memory are: (i) Total Memory Usage in GB (ii) Total Memory Usage in Percentage (iii) Working Set of profiled process in GB and (iv) Peak Working Set of profiled process in GB. For CPU I am logging: (i) CPU usage in Percentage (ii) CPU Clock in MHz and (iii) Core wise CPU Usage in Percentage. This log is generated when a process is launched from my C# executable and it completes successfully. It is stored as a comma separated value (.csv) file that is passed to R for plotting graphs.

2.2. PLOTTING

For visualization I have used R along with Shiny dashboard and Plotly. Using Shiny in R we can create web interfaces that can take user inputs and generate reactive outputs accordingly. Plotly is famous javascript API available for many platforms that enables plotting in Shiny. Plotly for R is open source. CSV log generated from C# executable is uploaded to R and then you can start processing it for plotting graphs. The file is then processed in R and three plots are generated: (i) CPU vs Memory usage in % (ii) CPU vs Cores usage in % and (iii) System/Total vs Process Memory usage in GB also the file is displayed as a table in the web interface.

2.3. HOW TO RUN

C# executable – open (double click) on the exe

R app – to host shinyapp run this command: runApp("C:\\Path\\to\\app.R\\file", host = getOption("shiny.host", "0.0.0.0"))

2.4. DOWNLOADS

1. C# executable - http://www.mediafire.com/download/6d6og6u0o83tdh6/perfloggengui_exe.zip
2. C# code - http://www.mediafire.com/download/h930k9biya6r4hh/MainForm.cs
3. C# project - http://www.mediafire.com/download/2fhqhye7zzm24ih/perfloggengui_project.zip
4. R app/code - http://www.mediafire.com/download/r344mpzdx4gq63s/app.R
5. Outlook.csv - http://www.mediafire.com/download/81dacva4jzlnvps/outlook.csv

6. Myperflog.csv - http://www.mediafire.com/download/s1pa7f2uo0mj3g0/myPerflog1_CPUMEM.csv

3. SCREENSHOTS

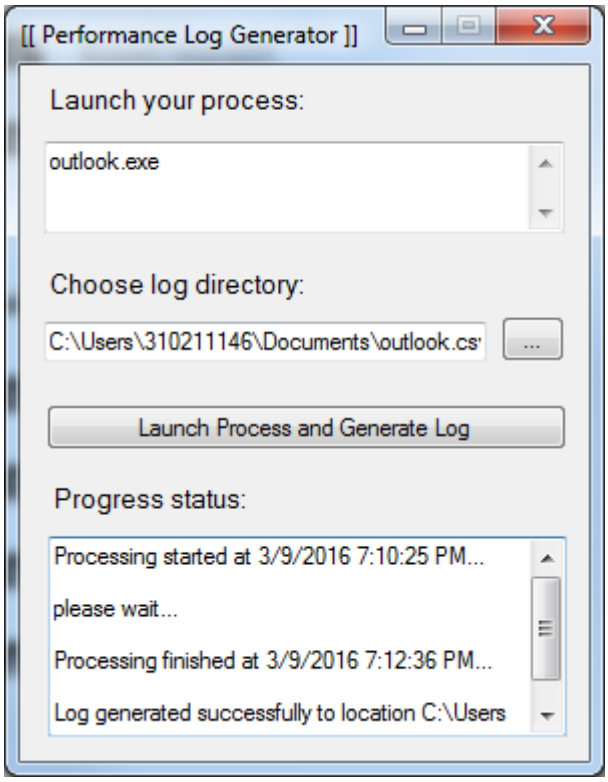

Fig -1: Log generator in C#

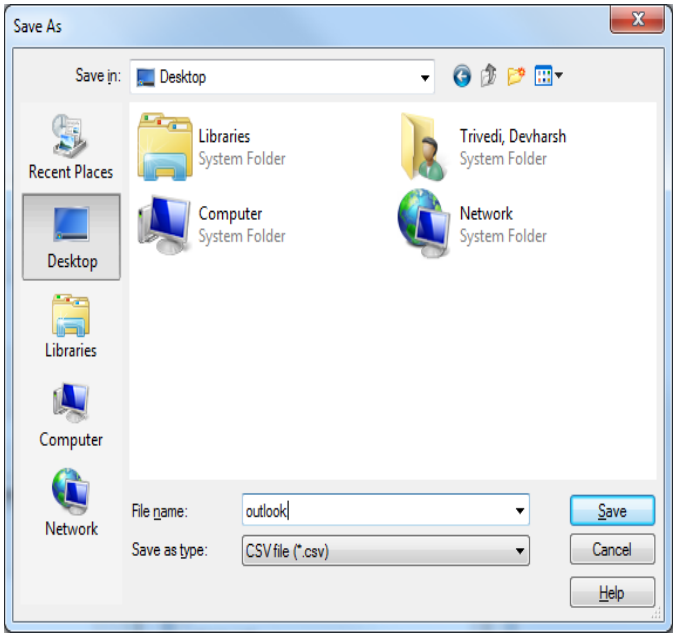

Fig -2: Choose an output file

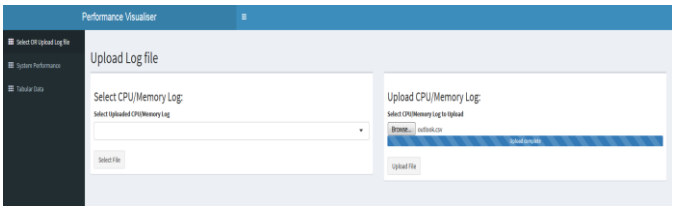

Fig -3: Select or upload a log file

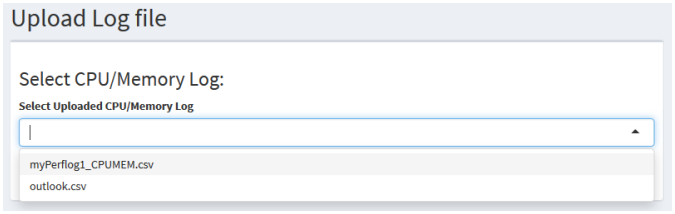

Fig -4: Log file dropdown menu

Fig -5: CPU/Mem usage plots

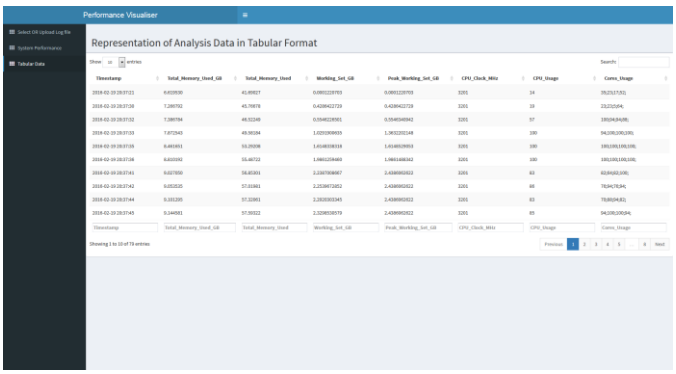

Fig -6: Log file in tabular format

```
> summary(cpudata)
   Timestamp                 Total_Memory_Used_GB Total_Memory_Used Working_Set_GB      Peak_Working_Set_GB CPU_Clock_MHz
 Min.   :2016-03-09 11:51:49 Min.   :4.402        Min.   :27.54     Min.   :0.0001488 Min.   :0.0001488    Min.   :1580
 1st Qu.:2016-03-09 11:52:09 1st Qu.:4.509        1st Qu.:28.21     1st Qu.:0.1207828 1st Qu.:0.1218662    1st Qu.:1580
 Median :2016-03-09 11:52:28 Median :4.514        Median :28.24     Median :0.1274757 Median :0.1326790    Median :2371
 Mean   :2016-03-09 11:52:28 Mean   :4.506        Mean   :28.19     Mean   :0.1190331 Mean   :0.1222087    Mean   :2028
 3rd Qu.:2016-03-09 11:52:48 3rd Qu.:4.515        3rd Qu.:28.25     3rd Qu.:0.1302986 3rd Qu.:0.1352405    3rd Qu.:2395
 Max.   :2016-03-09 11:53:07 Max.   :4.520        Max.   :28.28     Max.   :0.1322746 Max.   :0.1352425    Max.   :2395

   CPU_Usage        Cores_Usage
 Min.   : 0.00   0;0;0;0;  : 4
 1st Qu.: 6.00   0;6;6;0;  : 2
 Median :11.00   6;0;0;0;  : 2
 Mean   :15.49   6;0;18;6; : 2
 3rd Qu.:20.50   6;12;12;0;: 2
 Max.   :87.00   0;0;12;0; : 1
                 (Other)   :58
```

Fig -7: Summary of log data

4. WORKFLOW

Step 1: Open C# executable and enter command to launch the process that you want to profile. In this case I am profiling outlook.exe for demo purpose. You can enter any exe with its full path to launch that exe and if you have added its path to PATH environment variable then you can call that exe directly.

Also choose a valid directory and appropriate filename where you want to save the generated log file for your convenience. You don't need to append .csv to your file name as it will be added automatically.

Then simply click on the button "Launch Process and Generate Log" and it will generate log and save csv to defined location. You can also see the progress in textbox below it.

Step 2: After successfully generating log file go to R web interface where you can upload your latest generated log file via "Upload File" button.

Step 3: After uploading the file you can view it under select uploaded CPU/Memory log dropdown. Choose your log file to visualize and click "Select File" button.

Step 4: You will be automatically redirected to System Performance tab where you can find multiple plots for CPU, Memory and Cores usage.

Step 5: You can also see the content of log file in Tabular Data tab and filter it with individual search boxes provided along with them at bottom of each column.

You can also filter multiple columns at a time. Also you can go to your R console and perform any operation on log that is stored under cpudata variable. For example you can do summary(cpudata) which will show a summary for that dataset.

5. CONCLUSION

You can use these applications to evaluate performance of your application. You can modify and redistribute as per your requirement. If this helps you then you can cite this paper in your research work.

ACKNOWLEDGEMENT

I would like to thank my industry project guide Mr. Manish Kumar (MR-Recon, Philips Innovation Campus, Bangalore) for guiding and helping me throughout the course of visualization of data in R.

REFERENCES

[1] https://www.microsoft.com/en-in/download/details.aspx?id=42642
[2] http://www.icsharpcode.net/OpenSource/SD/Download/#SharpDevelop5x
[3] https://cloud.r-project.org/bin/windows/base/
[4] https://www.rstudio.com/products/rstudio/download/
[5] http://shiny.rstudio.com/tutorial/
[6] https://rstudio.github.io/shinydashboard/
[7] https://github.com/daattali/shinyjs
[8] https://plot.ly/r/

BIOGRAPHY

Devharsh Trivedi has done B.E. in Computer Engineering from Gujarat Technological University and now pursuing M.Tech. in Computer Science and Engineering – Information and Network Security from Nirma University.