

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

LOGGING 2.1.

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/6d6og6u0 o83tdh6/perfloggengui exe.zip
- 2. C# code http://www.mediafire.com/download/h930k9biy a6r4hh/MainForm.cs
- 3. C# project http://www.mediafire.com/download/2fhqhye7z zm24ih/perfloggengui project.zip
- 4. R app/code http://www.mediafire.com/download/r344mpzd x4gq63s/app.R
- 5. Outlook.csv http://www.mediafire.com/download/81dacva4j zlnvps/outlook.csv

IRJET

e-ISSN: 2395 -0056 p-ISSN: 2395-0072

6. Myperflog.csv

http://www.mediafire.com/download/s1pa7f2uo 0mj3g0/myPerflog1_CPUMEM.csv

3. SCREENSHOTS

[[Performance Log Generator]]	
	Launch your process:	
	outlook.exe	
	Choose log directory:	
	C:\Users\310211146\Documents\outlook.cs'	
	Launch Process and Generate Log	
	Progress status:	
	Processing started at 3/9/2016 7:10:25 PM	
	please wait	
	Processing finished at 3/9/2016 7:12:36 PM	
	Log generated successfully to location C:\Users 👻	
L		

Fig -1: Log generator in C#



Backet Topket right Bocket Topket right Bocket Topket right Bocket Topket right Bocket Topket right Select CPUI Memory Log: Select CPUI Memor		Performance Visualiser	•	
E Tourison Select CPU/Memory Log: Mentiplaado (POliwwoy Lig Mentiplaado (POliwowo) (Poliwy Lig Mentiplaado (POliwowo) (Poliwy Lig Mentiplaado (POliwowo) (Poliwy Lig Mentiplaado (Poliwowo) (Poliwy Lig Mentiplaado (Poliwowo) (Poliwowo) (Poliwy Lig Mentiplaado (Poliwowo) (Poliwo) (Poliwowo) (Poliwowo) (Poliwowo) (Poliwowo) (Pol	🖬 Select OR Upload Log Ne 📰 System Performance	Upload Log file		
Prec. and on Lideote	🗮 Tabular Data	Select CPU/Memory Log: Select Upluaded CPU/Memory Log		Upload CPU/Memory Log: Select ON/Memory Log II Upload
upinal fie		Select File	•	Essec schnictr galardinate Uptad Fie

Fig -3: Select or upload a log file

Upload Log file	
Select CPU/Memory Log: Select Uploaded CPU/Memory Log	
nyPerflog1_CPUMEM.csv outlook.csv	

Fig -4: Log file dropdown menu





	remormance visualiser									
Select OR Upload Log file	Representation of Analysis Data in Tabular Format									
	Shew 10 entries							Search		
	Terestamp	Total_Homory_Used_GB (Total_Hemory_Used	Working_Set_GB	Peak_Working_Set_GB	CPU_Clack_Mitz	CPU_Usage 0	Cores_Usage		
	2018-02-29 20 31/23	64029530	41.49427	0.0001220703	0.0001228/03	3201	34	39(23)(17)(52)		
	2016-02-19 20:37:30	1,296792	45.75870	0.4205422729	0.4389422729	3204	29	23(23)5(4)		
	2818-02-29 28:37:52	1.385784	45.52349	0.5548228505	0.5546348942	3200	57	10(14)44(0)		
	2818-02-29 28 37 33	1.872543	45.38354	1.0202000433	1.3632202148	3201	190	94,000,000,0001		
	2818-02-29 28 37 35	8-441451	53.29208	1.6148338338	1.6348529053	3294	330	380,100,100,100;		
	3016-02-09303736	8.800292	55.48722	1.9802258460	1.9851498342	3264	130	130,100,100,100;		
	2816-02-09-28:27:41	6.827850	54.85301	2.3387008667	2.4380803822	3204	42	62,64(62,500)		
	3016-02-0930:37:42	9.052525	\$7.01941	2.3539472452	2.43809(2)(22	3264	85	TER/TER:		
	2018-02-29 20:37/44	9.311295	\$7,32963	2.2828303345	2,4386802822	3204	10	78(81(94)82)		
	2018-02-29 20:37545	9.344581	57.99322	2.3298538579	2,43869(2)(22	3200	85	9420820854		
	Timestamp	Total_Memory_Used_GB	Total_Memory_Used	Working_Set_SB	Peak, Worklog, Set, 68	CP9_Clock_MHz	CPU_Usage	Cares_Usage		
	Showing 1 to 10 of 79 entries						Previous 1 2 3	4 5 8 Not		

Fig -6: Log file in tabular format

> summary(cpuda	ta)										
Timestamp		Total_M	temory_Used_GB	Total_	<pre>/emory_Used</pre>	working	g_Set_GB	Peak_W	orking_Set_GB	CPU_Clo	ick_MH
Min. :2016-0	3-09 11:51:49	Min.	:4.402	Min.	:27.54	Min.	:0.0001488	Min.	:0.0001488	Min.	:1580
1st Qu.:2016-0	3-09 11:52:09	1st Qu.	:4.509	1st Qu.	:28.21	1st Qu.	:0.1207828	1st Qu.	:0.1218662	1st Qu.	:1580
Median :2016-0	3-09 11:52:28	Median	:4.514	Median	:28.24	Median	:0.1274757	Median	:0.1326790	Median	:2371
Mean :2016-0	3-09 11:52:28	Mean	:4.506	Mean	:28.19	Mean	:0.1190331	Mean	:0.1222087	Mean	:2028
3rd Qu.:2016-0	3-09 11:52:48	3rd Qu.	:4.515	3rd Qu.	:28.25	3rd Qu.	:0.1302986	3rd Qu.	:0.1352405	3rd Qu.	:2395
Max. :2016-0	3-09 11:53:07	Max.	:4.520	Max.	:28.28	Max.	:0.1322746	Max.	:0.1352425	Max.	:2395
CPU Usage	Cores Usa	ae									
Min. : 0.00	0:0:0:0: : 4										
1st Ou.: 6.00	0:6:6:0: : 2										
Median :11.00	6:0:0:0: : 2										
Mean :15.49	6:0:18:6: : 2										
3rd Ou.:20.50	6:12:12:0:: 2										
Max. :87.00	0:0:12:0: : 1										
	(Other) :58										
	-		<u> </u>								
Fig -7: 3	Summa	ry o	ot log d	ata							

Fig -2: Choose an output file

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

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.