ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 6, June 2017

Machine Learning Algorithm for Process Performance Testing

Bharati Nadar[1], Mohammed Saadan Khan[2], Ameya Parkar[3]

P.G. Student, Department of M.C.A., VES Institute of Technology, Mumbai, Maharashtra, India [1,2]

Assistant Professor, Department of M.C.A., VES Institute of Technology, Mumbai, Maharashtra, India [3]

ABSTRACT: When application components are deployed across multiple servers then any change or upgrade in the application would affect the behavior of these components. Performance testing of these components is essential for the organization. The process of testing the performance can be automated using a machine learning algorithm. Using this algorithm the overall purpose of testing can be achieved with less or no human intervention as it learns from the historical data and generates a summarized Html email report notifying the behaviour of the process to the developer and tester, using which any major business decisions can be made if there is any major deviation in the performance.

KEYWORDS: Machine Learning, Process Performance Testing, Process Monitoring.

I. INTRODUCTION

Performance Testing is essential for providing stakeholders with information regarding speed, scalability and stability of their application. Moreover, performance testing helps us to uncover what needs to be improved before the product goes to the market. Without proper testing, our application is bound to suffer from issues like running slow when used among multiple users, inconsistencies and poor usability which would lead to bad reputation of the enterprise when the application is released in the market.

Process Performance testing plays a very important role for any application of an enterprise. Whenever we work on any application there are multiple processes that are invoked or are running in the background. If the developer changes or adds any new functionality to the application then it is possible that the behaviour of the process changes accordingly. This deviation in the process performance can sometimes be as expected or sometimes it may act differently. To monitor and test this process performance becomes a tedious task to a tester when he has to do it manually. Imagine if the process runs across several servers then monitoring and keeping track of the behaviour of each process will be tiresome and what if there are multiple processes that need to be monitored?

II. RELATED WORK

This testing and monitoring task of various processes across the servers can be achieved using a machine learning algorithm. Machine learning is a sub-type of artificial intelligence because it allows computers to get into a mode of self-learning without being explicitly programmed. When encountered with new data, the algorithm should be able to learn, evolve, enhance and execute smartly.

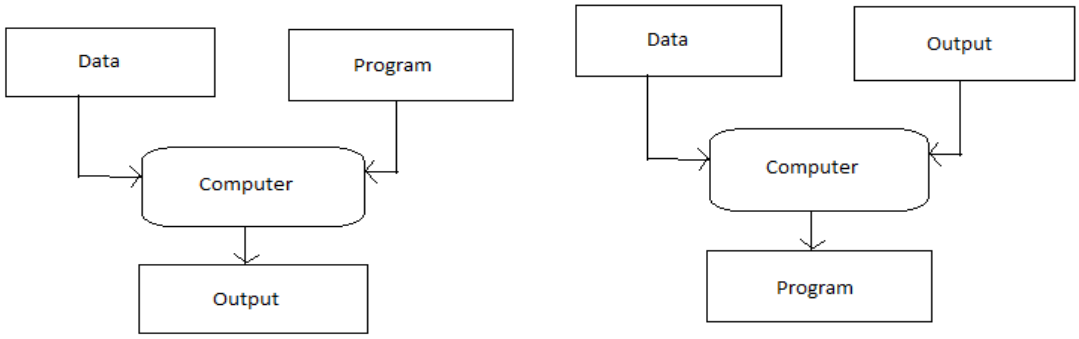
ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 6, June 2017



(a) (b)

Fig 1:- (a) Traditional Programming v/s (b) Machine Learning

In Traditional Programming, Input data was given to the computer and it generates the specified output. This means that the behaviour of the output depends on the program and the given data. While in machine learning input data along with the previous output is given and the program generates the new output accordingly i.e. the new output depends on the input data and the historical output data.

This concept can be implemented in the process performance testing to store the behaviour of multiple processes over the time period and use that stored data to determine the current behaviour change.

III. ARCHITECTURE

The proposed algorithm will read the list of server from the input file and then the program will connect to the server and it gathers all the process details that is running in that server. The program will then update the master excel file with the new data. The program learns from the previous data i.e the data stored in the excel sheet from the previous execution and generates the new output and stores that into the same excel file. The program can be written in any language which is best suitable for the test case and provides better result.

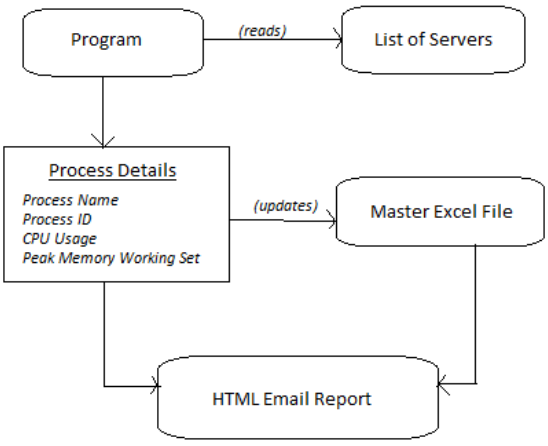


Fig 2:-Architecture of the Process Performance testing module

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 6, June 2017

As shown in the above figure, The program updates the master excel file and takes the data from it to create an email to notify the user of the completion of the program and provides the user with a summarized HTML Table Report.

IV. RESULTS

The main idea of the algorithm is that it will first connect with the server whose processes we need to monitor. Then it will collect the details of the intended processes like Process Name, Process Id, Process CPU Usage and Process Peak Memory Working Set.

A. Process Name: Name of the Process which is monitored.
B. Process ID: Unique Identification number to identify the instance of a process.
C. Process CPU Usage: Total CPU utilization of that particular process.
D. Process Peak Memory Working Set: This is the maximum memory used by the process so far.

Process Peak Memory Working Set (PMWS) places a very important role in this performance testing algorithm because this will be used as the historical data. After collecting all the information the program will store these data into a master excel file. The excel file will store the name of the server on which the processes are running, the name of the process, and the peak memory working set for the current date. When the program runs for the next day then a new column will be added with that date and the memory size of a particular process running on a particular server will be updated. Suppose if a process is not running in that server than the cell will be left blank.

Therefore when the program runs it will check if the master excel file exists or not. If not then the program will create it with the server name as the first column, the name of the process to be monitored on it as the second column and the peak memory working set (in MB) as the third column. Now when the program runs the next time then new column will be added and the memory usage of that particular process will be updated. If suppose on some day a new process is running on any server than that detail will be automatically added to the new row of the first and second column and its corresponding column for peak working set will be updated.

	A	B	C	D	E	F
1	Server	Process Name	Peak Memory (10/05/2017)	Peak Memory (11/05/2017)	Peak Memory (12/05/2017)	
2	server1	service1.exe	1	1.2	1	
3	server1	service2.exe	2	2.3	1.9	
4	server2	Application1.exe	3.5	3.8	4.5	
5	server3	Application2.exe	2		2	
6	server1	Application1.exe		3	3.2	
7	server4	service1.exe	3			
8						
9						

Fig 3:- Master Excel File

Once the new data is updated in the master excel sheet, the algorithm will generate an email HTML table report which will contain the list of processes and its average memory usage until the day before and its current peak memory working set. The average memory usage is calculated from the master excel sheet where the code uses the historical data to calculate the average memory usage of a particular process in a particular server.

The email will contain the HTML table which would contain a summary report of the process performance test. The algorithm will further calculate and highlight the process whose current memory usage exceeds the average memory usage. Suppose the threshold value is set to 0.5MB then all the process whose current peak memory usage is greater than the average memory usage value by 0.5 will the highlighted and will be listed at the top of table followed by the remaining processes in the master excel sheet.

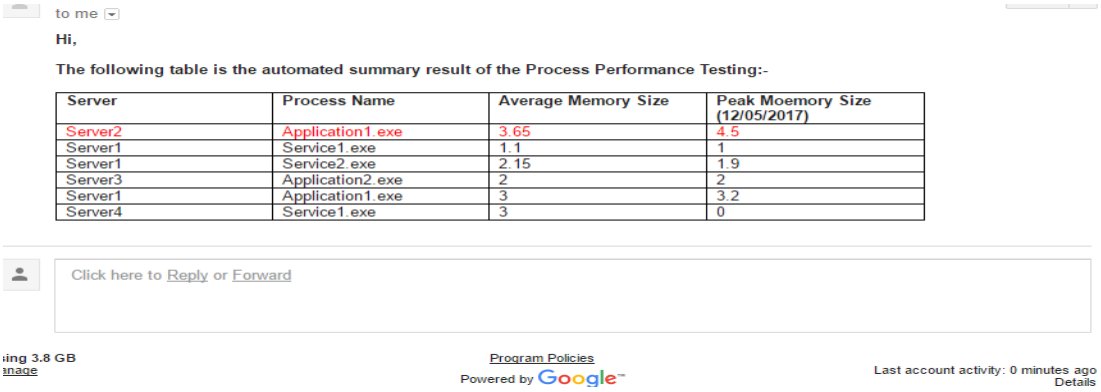
ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 6, June 2017



to me

Hi,

The following table is the automated summary result of the Process Performance Testing:-

Server	Process Name	Average Memory Size	Peak Moemory Size (12/05/2017)
Server2	Application1.exe	3.65	4.5
Server1	Service1.exe	1.1	1
Server1	Service2.exe	2.15	1.9
Server3	Application2.exe	2	2
Server1	Application1.exe	3	3.2
Server4	Service1.exe	3	0

Click here to Reply or Forward

ing 3.8 GB
nage

Program Policies
Powered by Google

Last account activity: 0 minutes ago
Details

Fig 4:- Email Test Result

Advantages:

There are several advantages of this algorithm and some of the benefits are listed below:-

1. This process performance testing algorithm will show the developer the memory usage of a particular process across the servers with the help of which the developer can keep track of the change in memory usage of any process after any update in the application.
2. The email report will highlight the process which uses more memory on that particular day than its average working set thus easing the monitoring and tracking process of a tester.
3. It eliminates the effort required by the tester to monitor each and every process across multiple servers. It automatically stores the data for future use and generates the email report to notify the tester. The tester has to just schedule the script to run on a daily basis and the report will be generated thus completely eliminating human intervention.
4. The algorithm will take the previous output and smartly proceed with its execution and generates the result and stores the new output for future processing.
5. The user can use the excel sheet to generate any type of graph to view the behaviour of any process over the period of time. If any process takes more memory after a period of time and continues to do so then any business decisions can be taken over it to improve its performance.

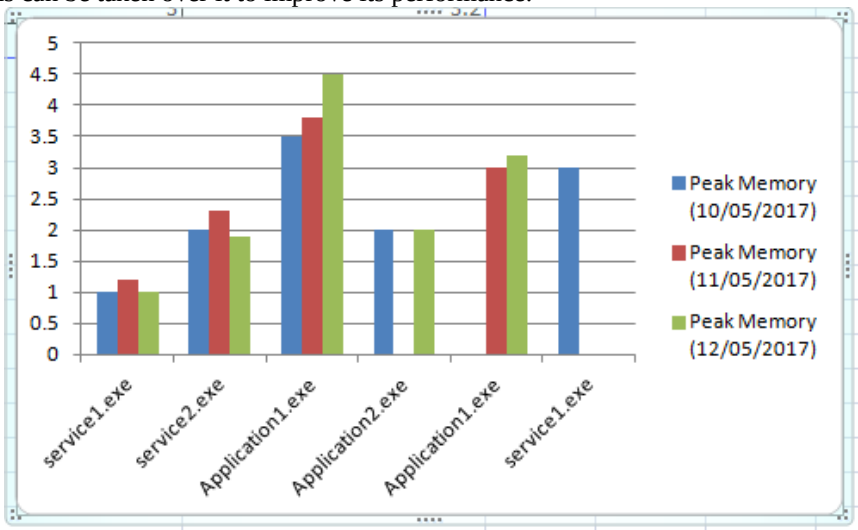


Fig 5:- Bar Graph showing the memory usage of the process

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 6, June 2017

V. CONCLUSION

Incorporating machine learning algorithms into any type of testing will always be beneficial for the tester as well as developer to minimize the effort and human errors in monitoring and it helps to deliver a better throughput. This algorithm will allow the user to have a program which would require less human intervention and provides a very efficient result which would be stored and further used. Any programming language can be used to develop this algorithm which the developer finds efficient and robust enough to handle any exception conditions.

REFERENCES

1. Kevin R Murphy, Machine Learning: A Probabilistic Perspective, 2012.
2. Shai Ben-David and Shai Shalev-Shwartz, Understanding Machine Learning: From Theory to Algorithms, 2014.
3. Ms. S. Sharmila and Dr. E. Ramadevi, Analysis of Performance Testing on Web Applications, International Journal of Advanced Research in Computer and Communication Engineering, ISSN (Online) : 2278-1021 ISSN (Print) : 2319-5940 Vol. 3, Issue 3, March 2014.
4. H. Sarojadevi, ―Performance Testing: Methodologies and Tools‖, Journal of Information Engineering and Applications, ISSN 2224- 5758 (print) ISSN 2224-896X (online) Vol 1, No.5, 2011.
5. http://www.guru99.com/performance-testing.html.
6. http://zenq.com/Portals/0/PerformanceTestingWhitePaperrevisedV1.1.pdf.
7. https://www.coursera.org/learn/machine-learningMuthukumar S, Dr.Krishnan .N, Pasupathi.P, Deepa. S, "Analysis of Image Inpainting Techniques with Exemplar, Poisson, Successive Elimination and 8 Pixel Neighborhood Methods", International Journal of Computer Applications (0975 – 8887), Volume 9, No.11, 2010.