INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)

e-ISSN : 2583-1062

Impact Factor : 5.725

www.ijprems.com
editor@ijprems.com

Vol. 03, Issue 11, November 2023, pp : 404-408

QUICK MB: A MICRO BENCHMARKING TOOL FOR LITE VIRTUAL MACHINES

Mr. Yugansh Garg[1], Ms. Bhavana Gupta[2]

[1]Assistant Professor, Department of Computer Science & Engineering School of Research & Technology People's University Bhopal, India.

[2]Assistant Professor, Department of Computer Science & Engineering, Sagar Institute of Science and Technology, Gandhinagar Bhopal, India.

ABSTRACT

Quick MB is a versatile microbenchmarking tool that can be used to assess the performance of virtual machines and docker containers, ranging from low-end 0.25 GB RAM machines to high-end 8 GB RAM machines with single or multiple cores. It supports both x86 and ARM based machines and can benchmark a wide range of Debian-based operating systems, including Ubuntu, Debian, Mint, BOSS, Lubuntu, and more. Unlike many other benchmarking tools that fail in ARM or command-line environments, QuickMB operates seamlessly on a variety of virtual hardware, from micro to medium-sized machines provided by various cloud providers.

Keywords: Benchmarking, Microbenchmarks, Performance analysis, Micro Virtual Machines, Virtualization, Cloud Computing

Code metadata

S.No	Code metadata description	Details
1.	Permanent link to code/repository used for this code version	https://github.com/yuganshgarg97/QuickMB.git
2.	Legal code license	GNU General Public License (GPL)
3.	Code versioning system used	git
4.	Software code languages, tools and services used	Python 3.6, Linux shell Script
5.	Compilation requirements, operating environments and dependencies	Ubuntu, Zorin OS, Debian OS, Linux Mint OS or any other Debian Based operating system.
6.	If available, link to developer documentation/manual	http://dx.doi.org/10.13140/RG.2.2.17571.73762
7.	Data Sets Generated using tool	https://github.com/yuganshgarg97/QuickMB-Data
8.	Support email for questions	yuganshgarg97@gmail.com

1. INTRODUCTION

QuickMB is a microbenchmarking tool that can be used to assess the performance of virtual machines and docker containers. It operates seamlessly on a variety of virtual hardware, ranging from low-end 0.25 GB RAM machines to high-end 8 GB RAM machines with single or multiple cores, provided by various cloud providers. The software is compatible with both x86 and ARM based processors.

QuickMB can benchmark a wide range of Debian-based operating systems, including Ubuntu, Debian, Mint, BOSS, Lubuntu, and many more. The software is coded in Python 3.6 and uses built-in modules such as os, time, csv, and pandas. The size of the software is 25 MB. This software requires a virtual or real machine with a low-end configuration of 0.25 GB RAM to a high-end configuration of 8 GB RAM. The machine can be hosted over any x86 or ARM based processor.

2. LITERATURE SURVEY

2.1 Microbenchmarks

Microbenchmarks deal with the smallest units of code and are a straightforward metric that measures the performance of a specific piece of code. Unlike regular benchmarking, which evaluates the runtime performance, microbenchmarks refer to a tiny piece of code and therefore run quickly. It is essential to make sure that microbenchmarks are used appropriately as implementing them when they are not necessary is a waste of time. The simplicity of microbenchmarking is both its greatest strength and weakness. This simplicity allows for the identification of performance issues by narrowing down the components and paths involved, but can also limit its usefulness. Before adding microbenchmarks to a project, it is crucial to validate their usefulness.[1], [2]

INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)

e-ISSN : 2583-1062

Impact Factor : 5.725

www.ijprems.com
editor@ijprems.com

Vol. 03, Issue 11, November 2023, pp : 404-408

2.2 Performance Analysis

System performance depends on the following factors such as CPU, GPU, memory and storage. Thus, performance analysis is done by giving the machine a certain set of tasks to calculate factors such as the time taken to complete the tasks, the CPU load during the task, and the main memory load used during the process.[3], [4]

Blesson Varghese used the 6 methodology which divided the test into the two categories i.e capture attributes and group attibutes. Capture attributes included the tests to find the computational ability of the machines where as the group attributes included the inter process communications, tranfers and similar tasks.[3]

Cloud assessment by application:

Virtual machines are then ranked according to their empirical performance (in this paper performance assessment is time-sensitive to complete execution). Values of individual criteria for

in $-\mu$ the evaluation performance is normalized using $\bar{v}_{i,j} = (V_{i,j} - \mu_j)/\sigma_j$ where μ_j is the mean value of $v_{i,j}$ on m VM and σ_j is standard deviation in i,j over m VM. Normalized values are used to evaluate Mpi Vms.[3]

Aaron Paradowsk while comparing the V.Ms hosted on Cloud stack and open stack took in account the factors including Processing speed, time required to process certain number of instructions and load on the Processor and Memory.[5]

Shrutika Dhargave in her paper titled "Evaluation of different Hypervisors Performances using Different Benchmarks" benchmarked hypervirsors using the application benchmarking methods including Hadoop Benchmark, SIGAR framework and GPU Pass-through Performance, FTP and HTTP approache. [6]

This method compares the performance of hypervisors but does not compare the quality of service providers as a whole system.

The paper titled "Experimenting with Application-Based Benchmarks on Different Cloud Providers via a Multi-cloud Execution and Modeling Framework" presents a wider picture of cloud perfromance analysis by different service providers. It compares the cloud service providers on the basis of the performnce against the set of applications. It applies the The process is described in order to achieve an optimal compromise between the parameters. Although this work is more advanced in the field of combined metric investigation, yes .[7]

CloudCmp provides a methodology and has a goal very similar to our approach to estimate the performance and cost of legacy cloud-deployed applications. Strong A tial cloud customer can use the results to compare different providers and decide whether should migrate to the cloud and which cloud provider is the best fit for their application them. CloudCmp identifies a set of performance metrics relevant to application performance- ance and cost, develop a benchmarking job for each metric, run jobs on different providers and compare.[7]

Papaer titled "Cloud Service Benchmarking " by David Bermbach presents a client side prosepective of cloud benchmarking and performance comparision however the method implemented by David is also the application based benchmarking, which is focussed on the high end machines, the entry level machines were ignored as usual.[8]

3. PROBLEM STATEMENT

1. Majority of the tools such as Geekbench, Cinebench often crash while testing on small machines. Majority of the above-mentioned tools execute workloads such as rendering, ray tracing or some other A.I based algorithms, but in case of micro virtual machines, these set of complex commands cannot be executed, thus either the program stops or the machine crashes.[9]
2. Majority of the benchmarking researchers use application-based bench marking. In this case, If the application used for bench marking gets a version update, the results of benchmarking may vary.

IMPLEMENTATION

To benchmark the light Machines The tool performs the set of tasks on the machines and monitors the three main attributes.

1. Time elapsed during the task
2. Average Percentage of Processor utilized during the task
3. Average Percentage of memory utilized during the task

Based on the previous Benchmarking done we divided the tasks in two categories as described in figure 1.

1. Data Processing Strength
2. Computational Strength

INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)

www.ijprems.com
editor@ijprems.com

Vol. 03, Issue 11, November 2023, pp : 404-408

e-ISSN : 2583-1062

Impact Factor : 5.725

Table 1 : Description of Tests

Group 1	Data Processing Strength	Data Transferring Strength	1.Read / Write Test 2. Copy Test
		Data Encoding/ Decoding Strength	1. Compression Test 2. Extraction Test
Group 2	Computational Strength	Instruction Handling Strength	1. Encryption 2. Decryption 3. Hashing
		Arithmetic Calculation Strength	1.Random Number Generation Test 2. Sorting Test 3.Integer Handling Test
		Float Calculation Strength	1. Floating Point Test

a. Group 1 : Data Processing Strength-

This group of processes includes the processes that measures the ability of a machine to handle the basic operations on data including read write operations, copy operation, compression and extraction operation on the data.

- **Data Transferring Strength**

  Here we used a 1 MB file, and created 1024 copies of it. It performed 1024 read and write operations continuously. Secondly we copied the same file i.e 1 GB Data, from one folder to another. These tests were used to find the ability of the machine to handle the read, write and copy instruction.

- **Data Encoding/ Decoding Strength**

  We used 1 GB Data and performed the compression and extraction operations over it. During the operation we measured the time taken by the machine, the load on the processor and load on the memory.

b. Group 2: Computational Strength

This group is further divided into three set of tests which includes Instruction handling ability, Arithmetic calculation ability and floating-point handling ability.

- **Instruction Handling Strength**

  This set included three tests, namely hashing, encryption and decryption. In this Test we used a 1 GB iso image and performed the above operations over it.

- **Arithmetic Calculation Strength**

  This set of tests included the operations like Sorting, finding random integer values in a given range and calculating the prime numbers in the certain range. This Test puts the load on memory as well as Processor. This test ultimately test the calculation ability of a machine.

- **Float Calculation Strength**

  In this test the machines were put under the test to perform operation over the floating points. This included operations of addition, multiplication of floating-point numbers.

c. Implementation Algorithm Of Software

Following is the order of execution of tests on the basis of which the tests are conducted and results are generated.

Step 1: Run Setup Script

Step 2 Run Read Write Test Function-Script

Step 3: Run Copy Test Function-Script

Step 4: Run Compression Test Function-Script

Step 5: Run Extraction Test Function-Script

Step 6: Download an ISO Image for further Test

Step 7: Run Hashing Test Function-Script on ISO image

Step 8: Run Encryption Test Function-Script on ISO image

Step 9: Run Decryption Test Function-Script on ISO image

Step 10: Run Sorting Test Function-Script on a list of 100000 integers

Step 11: Run Random Number Generation Test Function-Script

Step 12: Run Integer Test Function-Script

Step 13: Run Floating Test Function-Script

Step 14: Process the generated Data-Script

Step 15: Remove the files generated During Test

Step 16: Generate final Matrix files

Step 17: Exit()

4. RESULTS

The tool generates a matrix that summarizes the performance of resources and provides a detailed overview of the machine's hardware and software configuration.

The tool creates four files: Data.docx, Matrix.csv, Matrix2.csv, and Matrix3.csv. Data.docx includes details on the hardware description and the start and end times of a specific test. Matrix.csv shows the performance of the machine in each test. Matrix2.csv displays the performance of the machine in each group of tests, and Matrix3.csv shows the performance of the machine in each subgroup of tests.

5. CONCLUSION

While majority of the benchmarking tools fail in testing the smaller machines and virtual machines the Quick-MB can benchmark the machines in this segments. Moreover it can also operate over a wide range of machines provided by different cloud vendors such AWS, MS Azure, Oracle cloud, GCP and IBM Cloud. While traditional benchmarking takes hours, Quick-MB can perform the benchmarking in minutes. Thus it fills the gap in the category of benchmarking tools and provides an option for the benchmarking the lite V.Ms. The tool can be used by the organizations which uses micro, small and medium V. Ms such as AWS t2 micro, GCP e2 and many more in the same segment for the purpose of testing strength of the virtual machines. The results generated can also be used to compare the above-mentioned V. Ms with V. Ms provided by the other cloud vendors. Mr. Garg in " Performance Analysis and Comparison of the Micro Virtual Machines Provided by the Top Cloud Vendors" used the same methodology for comapring the virtual machines belonging to the Public cloud service providers including GCP, IBM, Azure, Oracle and AWS.[9]

The tool can benchmark almost all the debian based operating systems including ubuntu, debian, Mint, BOSS, lubuntu and many more. While majority of the benchmarking tools crashes in the ARM or Cli based environments, the Quick - MB operates over almost virtual hardware in the category of micro, small and medium range of virtual machines provided by the different cloud vendors. The tool provides a comprehensive analysis of a machine's performance during a test. It produces results in the form of files that show information on operations, test duration, memory utilization, processor utilization, and swap utilization. The tool creates a report of the machine's performance every second during the test, including multiple files that display both task-specific and resource-specific details.

6. FUTURE SCOPE

The tool can be used by the organizations which uses micro, small and medium V.Ms such as AWS t2 micro, GCP e2 and many more in the same segment for the purpose of testing strength of the virtual machines. The results generated can also be used to compare the above mentioned V.Ms with V.Ms provided by the other cloud vendors. The tool is currently available on the debian based distros. In the future it can be developed for the Redhat based distros as well as windows OS. The results presented are in form of Matrix only, in future the results can displayed on web servers and can be represented in the graphical forms.

7. REFERENCES

[1] A. Iosup, S. Ostermann, N. Yigitbasi, R. Prodan, T. Fahringer, and D. Epema, "Performance analysis of cloud computing services for many-tasks scientific computing," IEEE Trans. Parallel Distrib. Syst., vol. 22, no. 6, pp. 931–945, 2011, doi: 10.1109/TPDS.2011.66.

[2] "What is microbenchmarking : | by Adservio, IT quality experts. | ADSERVIO | Medium." https://medium.com/adservio/what-is-microbenchmarking-8e6ea189da2d (accessed Oct. 18, 2022).

[3] B. Varghese, O. Akgun, I. Miguel, L. Thai, and A. Barker, "Cloud benchmarking for performance," Proc. Int. Conf. Cloud Comput. Technol. Sci. CloudCom, vol. 2015-Febru, no. February, pp. 535–540, 2015, doi: 10.1109/CloudCom.2014.28.

[4] A. Crape and L. Eeckhout, "A Rigorous Benchmarking and Performance Analysis Methodology for Python Workloads," in Proceedings - 2020 IEEE International Symposium on Workload Characterization, IISWC 2020, Oct. 2020, pp. 83–93. doi: 10.1109/IISWC50251.2020.00017.

[5] IEEE Computer Society et al., "Benchmarking the performance of openstack and cloudstack," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 7759 LNCS, no. February, pp. 173–188, 2015, doi: 10.1007/978-3-319-25414-2_14.

[6] S. Dhargave, "Evaluation of different Hypervisors Performances using Different Benchmarks," 2016. [Online]. Available: http://www.ijser.org

[7] A. Evangelinou et al., "Experimenting with application-based benchmarks on different cloud providers via a multi-cloud execution and modeling framework," Commun. Comput. Inf. Sci., vol. 512, pp. 213–227, 2015, doi: 10.1007/978-3-319-25414-2_14.

[8] D. Bermbach, E. Wittern, and S. Tai, Cloud service benchmarking: Measuring quality of cloud services from a client perspective. 2017. doi: 10.1007/978-3-319-55483-9.

[9] M. Y. Garg and M. B. Gupta, "Performance Analysis and Comparison of the Micro Virtual Machines Provided by the Top Cloud Vendors.," Int. J. Res. Publ. Rev., vol. 04, no. 02, pp. 1326–1333, 2023, doi: 10.55248/gengpi.2023.4229.