Environmental Sustainability Coding Techniques for Cloud Computing

Shakeel Ahmed
College of Computer Sciences and Information Technology
King Faisal University, Al-Ahsa, Saudi Arabia

***Abstract*—Cloud Computing (CC) has recently received substantial attention, as a promising approach for providing various information and communication technologies (ICT) services. Running enormously complicated and sophisticated software on the cloud requires more energy and most of the energy is wasted. It is required to explore opportunities to reduce emission of carbon in the CC environment that causes global warming. Global warming affects environment and it is required to lessen greenhouse gas emissions which can be achieved by adopting energy-efficient technologies that largely need to reduce the energy consumption caused during computations, for storing data and during communications to achieve Green computing. In literature, most of the energy-saving techniques focus on hardware aspects. Many improvements can be done in regard to energy efficiency from the software perspective by considering and paying attention to the energy consumption aspects of software for environmental sustainability. The aim of the research is to propose energy-aware profiling environmental sustainability techniques based on parameterized development technique to reduce the energy that has been spent and measure its efficiency inured to support software developers to enhance their code development process in terms of reducing energy. Experiments have been carried out and results prove that the suggested techniques have enabled in achieving energy consumption and achieve environmental sustainability.**

Keywords—Environmental sustainability; cloud computing; energy-efficient; software development life cycle; parameterized development; green computing

I. Introduction

Cloud Computing services comprised of large data centers, several virtualized server instances, networks with high band, and supporting systems for cooling and power supply. These equipment's can be classified into hardware and software and are accessed by remote users as shown in Fig. 1, and Cloud services are made accessible to the users through the usage of network equipment which connects the servers to the Internet which makes up the hardware equipment [1]. The user's applications/software managed by Cloud management system (CMS) are commonly referred as appliance and which runs on top of servers and is other supporting equipment includes power supply, cooling system, and data center building.

Green cloud computing became widely thought of issue in large organizations and they have targets to minimize the energy consumption and reduce unnecessary resources. Cloud computing is cost effective and highly scalable infrastructure for running High Performance Computing (HPC), enterprise and web applications. Moreover, there has been drastically growing demand of cloud infrastructure by different organizations which have resulted in the increased consumption of energy to run these data centers, which has become a serious concern. Where High energy consumptions increase the operational cost and decrease the profit margin of cloud providers. Further, this also leads to high carbon emissions which is not eco-friendly [2].

A. Energy Optimization in Cloud Computing

Energy efficiency can be well-defined as reduction of energy used for a given service or level of activity [4]. Nevertheless, owing to measure and convolution of data center equipment it is very difficult to describe service or bustle that could be observed for energy efficiency. Hence, [3] identified four scenarios within a system where energy is not used in well-organized way but it's lost or not used, as shown in Fig. 2.

Today, energy efficiency is the most vital constraints for cloud computing. It decides the operational costs and the benefits of capital investment. Since CC depend on data center industry deployed worldwide. Cloud applications optimization will not only reduces the data centers energy consumption, rather also will contribute in optimize overall cloud computing systems globally. In [5], issues are addressed in regards to reducing energy consumption of data centers, as well as in cloud computing. They include but not limited to virtual machines placement workload scheduling or load balancing as illustrated below:

1) Virtual machines placement: Virtualization is an efficient operation in cloud data centers. The physical equipment that is energy consumption can be replaced virtually in data center. VMs merging approaches try to accommodate lowermost conceivable number of physical machines to multitude a number of VMs. In virtualized environment, scheduling and load balancing are some techniques used effectively to manage operation in data centers. VM is ensuring efficient utilization of storage to serve hosted application workloads and then lead to minimize energy consumption in data center.

2) Workload scheduling: It contributes in improving the efficiency of data center in a way that specifies the resources that would be used to complete the work. The work could be data flow, processes or threads that are in turn waiting to being processed. The focus is to achieve efficient utilization of resources selected from the server with cost and energy effective. Additionally, energy efficient workload scheduling tends to ponder all active workloads onto a few numbers of servers with least possible communication resources.

3) Load balancing: The aim of load balancing is to optimize the resource usage, maximize the throughput and minimize response time, as well as to avoid overload for particular resource(s). Load balancing can efficiently distribute the incoming traffic among available servers in fast and reliable manner. It ensures that no one server has overloaded more than other, that may cause it to be down. As the inactive server (in case of shutdown) require energy to turn it on, and then heat is generated more and cooling system will consume electrical energy to keep good temperature. Load balancing is the important enabler for saving energy by avoiding unnecessary power consumption. However, recoveries after the server turned ON will causes some delays.

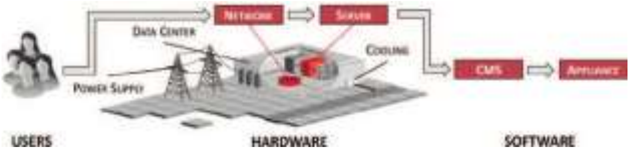


Fig. 1. Cloud Computing Data Center Domains [3].

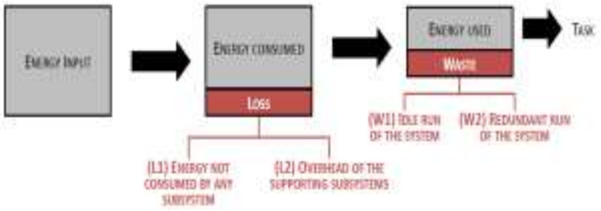

Energy Input
Energy consumed
Loss
Energy used
Waste
Task
(W1) Idle run of the system
(W2) Redundant run of the system
(L1) Energy not consumed by any subsystem
(L2) Overhead of the supporting subsystems

Fig. 2. Scenarios where Energy is Lost or Wasted within a System [3].

The structure of the article is as follows: Section 2 is related to the literature review and presents the discussion on the tools used by the developers for code development. Section 3 presents the framework of parameterized development phase. Section 4 presents the results, and Section 5 summarizes and concludes the paper.

II. Literature Review

There are quite many platforms/tools used by the developers in code development and deployment of Cloud based services and applications which include Windows Azure [6], Hadoop [7], Twister [8], Google App Engine [9], Microsoft Daytona [10] and Manjrasoft Aneka [11] in all these platform there is a lack of consideration for energy efficiency techniques to be followed by the developer. Although a general-purpose programing environment to rationalize and support designers and developers in creating energy-aware and energy-efficient applications is proposed in [12]. Most of the consideration in the literature has thorough research on to enhance the Cloud Computing energy efficiency and considering improved resource management to circumvent specific concerns like excessive power consumption and service-level agreements (SLA) violation reliability [13].

Consequently, various progresses have been presented like, Dynamic voltage and frequency scaling (DVFS) and Dynamic Power Management (DPM) methodologies used in monitoring the power consumption of servers according to the workload [14]. To optimize hosts virtual machine consolidation policies are used by migrating virtual machines (VMs) from one host VM to another VM [13]. Few models are proposed for improved forecast of the task consolidation in order to maximize resource utilization [15] power consumption for the servers [16]. A general outline/framework known as Mistral for optimizing physical hosts' power consumption [17], a CPU re-allocation algorithm which combines live migration techniques and DVFS and to cut the energy consumption and raise the performances in Cloud datacenters [18]. Though, there is a deficiency of research that grabs the concern of appropriately confirming energy-awareness from the design stage. Therefore, it is still a need for establishing the energy efficiency of Cloud infrastructures to increase a better understanding of energy efficiency and to nourish the policymaking at the service design stage. So, the applications or software must be prepared with energy efficient techniques at the implementation phase. This means that the developers need to design the software with Green Software Development Life Cycle (G-SDLC). Where the source code is studied by identifying and analyzing the hotspots, which causes energy consumption with help of tools or methods. Cloud computing and software development are the backbone to design software which is most energy efficient for cloud based applications.

As results, the researchers focus on the side of implementation of cloud components that affect energy consumption. Thus, components decide the type of cloud computing service models that are being used in the environment of cloud. Our effort in this research project is on software implementation by elaborating in energy optimization at software or application level. Focusing on energy consumption of applications' profiling at runtime and narrow down hotspots iteratively; to achieve G-SDLC.

Developers need to understand the estimation of the program energy consumption to avoid the power waste in early stage of development. The current research focuses on elucidate of green software development life cycle, especially in software implementation and parameterized methodologies to contribute in reducing energy consumption when it is running on cloud. The fact is that when the developers improve the source code with the help of parameterized phase; energy optimization of software can be achieved by determining the part of code that consumes high energy during the runtime in alignment with regular SDLC phases. Our enhancement is adding parameterized development phase which focus on code generated in implementation phase. As stated, the source code plays a vital role in achieving energy efficient software. Therefore, all efforts focused on unnecessary use of controls, libraries and some unnecessary instructions by applying the parameters to reduce the energy consumption of software when running in cloud environment.

III. PROPOSED SYSTEM

The aim of this work to propose and investigate the framework of software development life cycle that introduces a new phase called parameterized development phase to enhance the energy efficiency of the software as shown in Fig. 3, which helps the developers to optimize energy efficiency of software applications when it is being developed. Parameterized phase improves the software code by generating energy efficient code and to get efficient software which improves the source code for optimizing energy consumption and to determine the highest power consumption part in code. Therefore, the software application would be environment-friendly in cloud and same functionalities with other aspects of quality still maintained. This is of our interest in achieving and optimizing energy by implementing G-SDLC in cloud computing environment.

Parameterized phase provides guidelines and appropriate methodologies for developers that promote software's implementation phase. Set of parameters works as checklist that help in software code development. Such as data structure design, architectural design, abstract specification, and algorithm design. Applying the parameters should be based upon programming language used, application function and engineering hardware architecture. All these should be presented in earlier phase of software's specification and requirement. Thus, developers can modify those parts consuming the energy instead of re-writing the code from scratch. Energy optimization comes with best practice and experience; developers will be able to identify the part of the code demand a particular pattern to be modified.

Efficient code can be cost effective using different programming techniques that are used as parameters to adopt on source code. As result, developers can undertake code manually. Such as loop unrolling mechanism by executing the instructions within the loop in parallel. This will optimize compiler time as well as the energy required to compile the sequence loop instruction. For example, in a program that deletes 100 items a simple For loop can accomplish this task by calling the function delete( ). As a result of this adaptation, the enhancement with "loop unrolling" only 20 iterations of the jumps and conditional branches need to be run, instead of 100. Developers are responsible to choose loop control variables and number of operations inside the unrolled loop structure. So, the output is same as the original code produced.

A. Code Performance Measurement

The energy consumption depends on the application itself. If the code/application is extensively running with high usage of CPU and memory requirements then its execution will result in high energy consumption. Therefore, energy consumption here will be directly relational to the applications. Developer must put more efforts in regards to optimize the performance of code. Performance optimization is endless process. There is always an opportunity for the developer to improve and make the code run fast in efficient manner. Hence, software should be developed to be energy efficient either at earlier stage or during runtime.

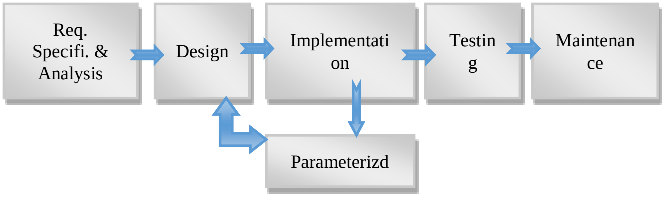


Fig. 3. Green Software Development Life Cycle.

The measurement of these parameters is the roadmap to know whether the software has good or low performance in terms of CPU utilization and memory it consumes. In other words, efficient application has good performance when its execution takes less time. However, inefficient application has low performance that leads to higher CPU and memory utilization, then higher energy consumption more time to execute Object-Oriented Programming Language most probably provides integrated garbage collection that helps the developer from dealing with memory de-allocation. However, garbage collection may consume resources in computation to take decision of which memory to free. Therefore, this could lead to decreased performance of the code [19]. Manual optimization remains mostly important at the source code level. Developer can use their experience and knowledge to perform optimization in creative manner to obtain good performance [20].

Optimizing the code will indicate good performance in regards to CPU and memory utilization. This can be measured by using performance profiler to identify code bottleneck. The code bottleneck can be found, for example, in loop that is executed hundreds/ thousands of times. Developer can redesign the body of loop in order to form in way that does not need to be executed hundreds/thousands of times [21].

B. Measure the Performance using Performance Profiler Tool in Microsoft Visual Studio 2017

Microsoft Visual Studio 2017 profiling tool, allows developer to analyze performance issues in their application code that is written with any programming languages of ASP.NET. It reports the performance from different perspective, including CPU usage, memory usage, performance wizard and others. It reports the performance of code by collecting information about the functions that performing work. Furthermore, provides timeline graph the developer can focus on precise sections of code [22]. Measuring the performance of code can be done using performance profiler in two steps: collect CPU/memory usage data and analyze CPU/memory usage data. To measure the time CPU is spending the CPU usage tool are used so that the developer can modify the code accordingly. Similarly, memory allocation shows the total memory allocated to execute the subject code [23] [24].

C. Performance Optimization Experiments using Proposed Parameter

One cannot really know about performance of code without measuring it. However, the function code of an application can be measured to see how long it takes and memory occupies. The good performance is defined by the function that taking less time to be executed with less memory utilization. On other hand, the low performance is defined by function that taking more time to be executed and consuming the system resources. The various Code Optimization techniques/parameters introduced in the earlier section are performed on two experiments discussed below by applying these parameters to observe how memory and CPU usage is optimized to enhance the source code and makes it performs efficiently. ASP.NET was used framework to develop website with traditional software development life cycle, and another version with G-SDLC. The reasons behind selecting ASP.NET are that, it is language independent in which support more than one programming language. It reduced the long lines of code. ASP.NET framework has magnificent features, in which gives alert for infinite loops, memory leaks and other wrong behaviors. It handles this immediately by killing the wrong behaviors and restarts them over again. This will help in increasing the performance and achieving efficient website/application. Microsoft Visual Studio 2017 is used to create these versions of website.

IV. Results

To demonstrate our proposed method, simple website with two versions which consists of four pages with similar design is developed. These websites run functions of reading around 400 records of data and inserts them into database in two different ways. Moreover, these records can be deleted from the database using two different methods. The database used is Microsoft Access 2010. Two different methodologies or functions on source code are used to deliver the same output. Performance measurements are made for the websites in regards to CPU usage and memory allocation using the performance profiler tool in Microsoft Visual Studio 2017. For simplicity, the website that needs to be enhanced is referred as website1. On other hand, the enhanced website is referred as website2. Both websites performed two functions insert and delete records from the database table. Performance assessment is performed on website1 regarding the insertion and deletion of records. Later, same assessment is performed on website2. In Website1, the insertion of records is done using for loop as follow:

```
For cnt = 0 To dt2.Rows.Count - 1
sql = "insert into ToEmployees (EmpName, EmpLog) values "
& _"('"& dt2.Rows(cnt).Item("SrcEmpName") & "' ,'" &
dt2.Rows(cnt).Item("SrcEmplog") & "')"
Try
conn.ConnectionString = ConStr
adapter.SelectCommand = New OleDbCommand(sql, conn)
adapter.Fill(dt)
Catch ex As System.Exception
Throw (ex)
Finally
conn.Close()
End Try
conn.Close()
```

Based on performance profiler tool, Fig. 4 indicates that the developer needs to implement some modification in order to improve the highlighted lines. These highlights can be accessed by investigating the peak point reported by CPU usage or running time addressed in Fig. 5 which shows that website1 executed with total of 1,792 samples collected as CPU usage to insert 400 records. The sampling is recommended method for starting the investigation of code and it is useful for finding issue of processor utilization [25].

Based on the performance profiler tool, the insertion function in Website1 is executed with total of 5,759 KB allocated as illustrated in Fig. 6. In this version of website1, the inserted records in the database table can be deleted using for loop.

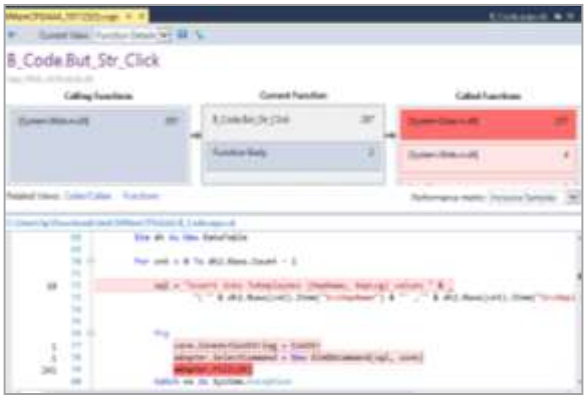
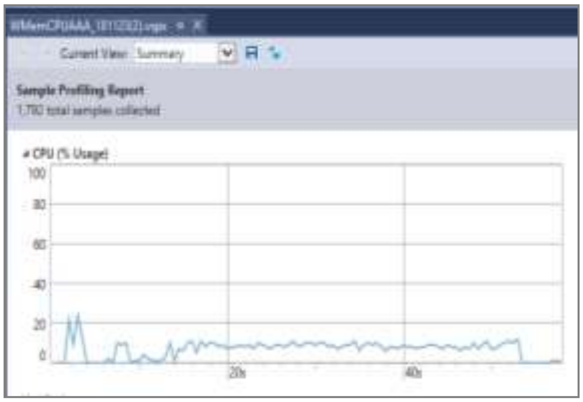

B_Code.But_Str_Click

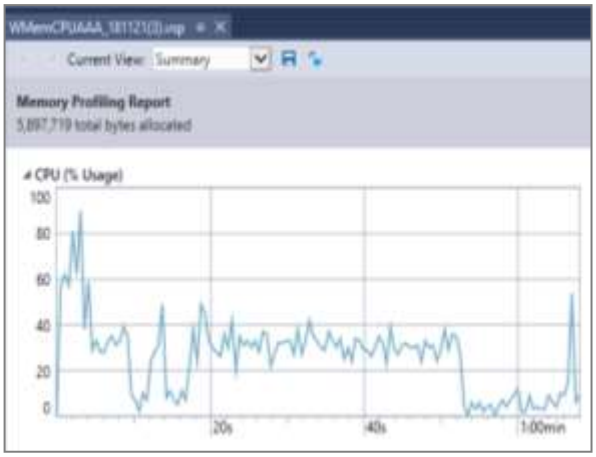
Fig. 4. Hot Spot Code of Inserting Records in Website1.


WMemCPUAAA_1811232).vspx

Current View: Summary

Sample Profiling Report
1,792 total samples collected

Fig. 5. Total CPU usage of Inserting Records in Website1.


WMemCPUAAA_181121(3).vsp

Current View: Summary

Memory Profiling Report
5,897,719 total bytes allocated

Fig. 6. Total Memory Allocation of Inserting Records in Website1.

```
For cnt = 0 To dt2.Rows.Count - 1
sql = "Delete from ToEmployees where EmpLog = '" &
dt2.Rows(cnt).Item("SrcEmpLog") & "'"
Dim adapter As New OleDbDataAdapter()
Dim conn As New OleDbConnection
Dim dt As New DataTable
Try conn.ConnectionString = ConStr
adapter.SelectCommand=NewOleDbCommand(sql,        conn)
adapter.Fill(dt)
Catch ex As System.Exception
Throw (ex)
Finally
conn.Close()
```

This function is executed with total of 4,358 samples collected as CPU usage to delete 400 records. Fig 7, is snapshot reported from the performance profiler tool. As this deletion function causes to release memory space, there was no result to report memory usage for this function, i.e. no need to locate memory.

The Website1 underwent modification after getting the performance assessment of it. This modification reflected on Website2. It uses methodology of inserting and deleting the data into database using single SQL command instead of having a loop. The following code is the modified part to insert records into database table:

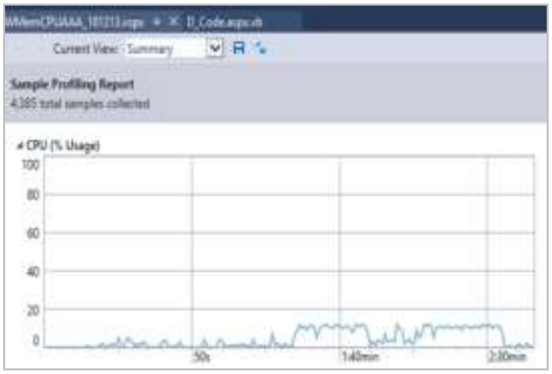
```
Dim sql As String = " INSERT INTO ToEmployees
(EmpName, EmpLog) SELECT SrcEmpName, SrcEmpLog
FROM(Employees) WHERE EmpChk= true"
adapter.SelectCommand = New OleDbCommand(sql, conn)
adapter.Fill(dt)
conn.Close()
```

Moreover, proper modification is applied in deletion function that causes deletion of the records that had been inserted into database table as follows:

```
Dim sql As String = "Delete * from ToEmployees "
 Dim adapter As New OleDbDataAdapter()
 Dim conn As New OleDbConnection
 Dim dt As New DataTable
Tryconn.ConnectionString = ConStr
adapter.SelectCommand=NewOleDbCommand(sql,    conn)
adapter.Fill(dt) Catch ex As System.Exception
 Throw (ex)
 Finally
conn.Close()
 End Try
```

Same procedures is followed to assess the performance in website1's followed by website2.Starting with insertion method; Fig. 8 indicates that the code is executed with total of 713 samples collected as CPU usage. This show an improvement of 60% in CPU performance compared to website1. This reflected the enhancement of insertion function that insert around 400 records in database table. This led to have faster execution in short duration as illustrated.

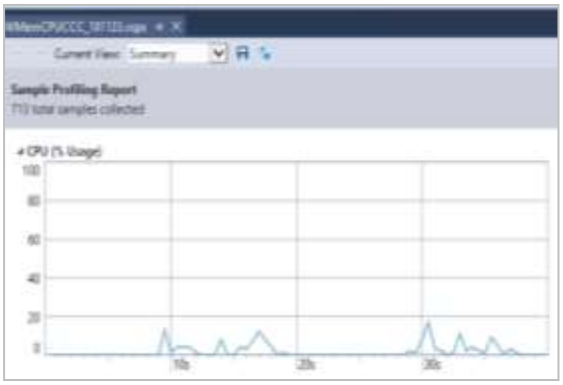
Fig. 9 shows total of 1,098 samples collected as CPU usage of delete command. This is an improvement of 74% of delete function compared to website1.



Current View: Summary

Sample Profiling Report
4,385 total samples collected

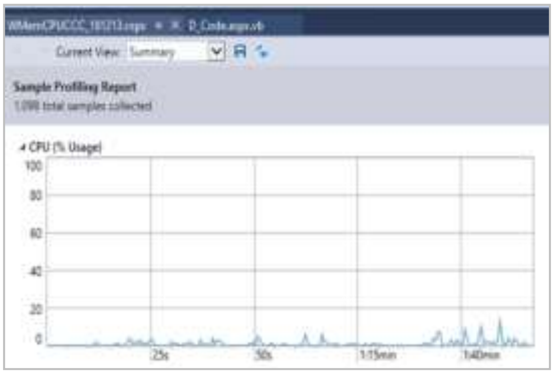
Fig. 7. Total CPU usage of Deleting Records in Website1.



Current View: Summary

Sample Profiling Report
713 total samples collected

Fig. 8. Total CPU usage of Inserting Records in Website2.



Current View: Summary

Sample Profiling Report
1,098 total samples collected

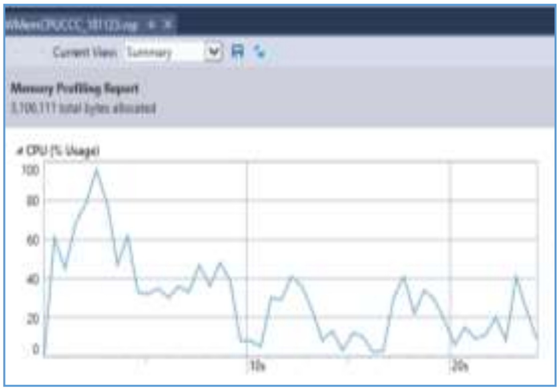
Fig. 9. Total CPU usage of Deleting Records in Website2.

Fig. 10 show that Website2 is executed with total of 3,033 KB allocated. This is an improvement of 47% in memory allocation compared to website1.

Fig. 11 summaries the result of performance assessment for the developed websites. This summary gives clear view of website2 improvement compared to website1 in regards to CPU usage and memory allocation. The insertion function of website1 is using for loop to populate database table with the selected records. On the other hand, website2 is using single SQL command to insert the records that meet the requirements with total improvement of 60% in regards to collected samples.

The reduction of total samples means less time of execution and response [26]. Fig. 12 illustrates the comparison of CPU usage for delete function in website 1 and website2 with total improvement of 74% in regards to collected samples.

Fig. 13 illustrates the improvement of total memory allocation of website1 and website 2by using appropriate methodology to insert data in database table. Memory allocation reduced by 47% in website2 compared to website1. Overall, this initial energy reduction will help in deliver energy efficient website when in it deployed in server.



WMemCPUCCC_181123.vsp

Current View: Summary

Memory Profiling Report
3,100,111 total bytes allocated

Fig. 10. Total Memory Allocation of Website2.

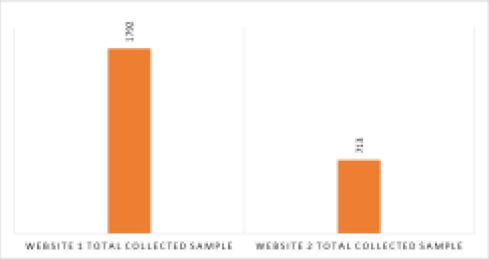


Fig. 11. CPU Collected Sample Improvement for Insert Function.

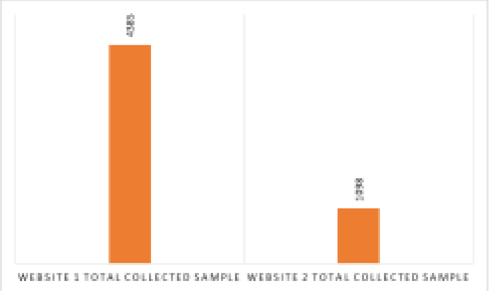


Fig. 12. CPU Collected Sample Improvement for Delete Function.

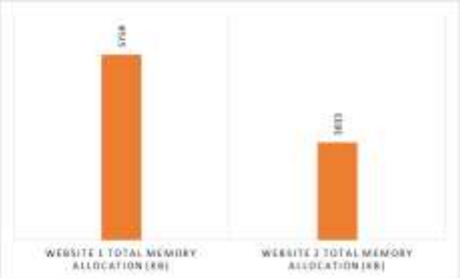


Fig. 13. CPU Collected Sample Improvement for Total Memory Allocation.

V. Conclusion

Energy efficiency ought to be the first priority of software or website development. A main concept is that developer must use his/her experience for any chance to optimize the source code. SDLC with the additional phase of parameters, produced energy efficient website in regards to CPU and memory usage. This optimization had been measured using performance profiler built-in tool in Microsoft Visual Studio 2017.The initial energy optimization version of the website provides perception of the website's performance for the developer before deploying it in server. This perception gives an opportunity to re-assess the performance, if website does not comply with the required criteria of having green cloud computing. Hence, the cloud energy consumption is controlled from software side as well as hardware side.

Software development improvement leads to reduce power costs. Therefore, getting green cloud and saving money. The latest processors have improved energy efficiency in hardware side. In software side, developer needs to contribute and follow appropriate parameters in implementation phase, as small change to software engineering can make a large difference in regards to energy and for environmental sustainability.

References

[1] AlNuaim, A. and Ahmed, S., 2018. Fog Computing: A Novel Approach to provide Security in Cloud Computing. Indian Journal of Science and Technology, 11(15), pp.1-6.

[2] Rajkumar Buyya, and Sukhpal Singh Gill. 2018. Sustainable Cloud Computing: Foundations and Future Directions. Business Technology & Digital Transformation Strategies, Cutter Consortium, 21, 6, 1-10.

[3] Mastelic, T., & Brandic, I. 2015. Recent Trends in Energy-Efficient Cloud Computing. IEEE Cloud Computing, 2, 40-47.

[4] M.M. Islam, M. Hasanuzzaman, Introduction to energy and sustainable development ,Energy for Sustainable Development, Academic Press, 2020, Pages 1-18, ISBN 9780128146453.

[5] Renugadevi, T.; Geetha, K.; Muthukumar, K.; Geem, Z.W. Optimized Energy Cost and Carbon Emission-Aware Virtual Machine Allocation in Sustainable Data Centers. Sustainability 2020, 12, 6383.

[6] Windows Azure: Microsoft's Cloud Platform — Cloud Hosting — Cloud Services. Available at http: //www.windowsazure.com/en-us/, 2020.

[7] Welcome to Apache, Hadoop! Available at http://hadoop.apache.org/, 2020.

[8] J. Ekanayake. Twister: Iterative MapReduce. Available at http://www.iterativemapreduce.org/, 2009.

[9] Google App Engine - Google Developers. Available at https://developers.google.com/, 2020.

[10] Daytona - Microsoft Research. Available at http://research. microsoft.com/en-us/projects/ daytona/, 2013.

[11] Manjrasoft Products Available at http://www.manjrasoft.com/ products.html, 2020.

[12] Xian, Y. Lu, and Z. Li. A programming environment with runtime energy characterization for energyaware applications. In Low Power Electronics and Design (ISLPED), 2007 ACM/IEEE International Symposium on, pages 141–146, 2007.

[13] Beloglazov, J. Abawajy, and R. Buyya. Energy-aware resource allocation heuristics for efficient management of data centers for Cloud computing. Future Generation Computer Systems, 28(5):755– 768, May 2012.

[14] Kliazovich, P. Bouvry, and S. Khan. DENS: Data Center Energy-Efficient Network-Aware Scheduling. In 2010 IEEE/ACM Int'l Conference on Green Computing and Communications & Int'l

Conference on Cyber, Physical and Social Computing, pages 69–75. IEEE, December 2010.

[15] R. Basmadjian, F. Niedermeier, and H. De Meer. Modelling and analysing the power consumption of idle servers. Sustainable Internet and ICT for Sustainability (SustainIT), 2012, pages 1–9, 2012.

[16] Y. Lee and A. Zomaya. Energy efficient utilization of resources in cloud computing systems. The Journal of Supercomputing, 60(2):268–280, March 2010.

[17] G. Jung, M. Hiltunen, K. Joshi, R. Schlichting, and C. Pu. Mistral: Dynamically Managing Power, Performance, and Adaptation Cost in Cloud Infrastructures. 2010 IEEE 30th International Conference on Distributed Computing Systems, pages 62–73, 2010.

[18] W. Chawarut and L. Woraphon. Energy-aware and real-time service management in cloud computing. In Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), 2013 10th International Conference on, pages 1–5, 2013.

[19] Matthew Hertz and Emery D. Berger, “Quantifying the Performance of Garbage Collection vs. Explicit Memory Management,” San Diego, California, USA,Rep. 1-59593-031-0/05/0010, October 2005.

[20] Rudrik Upadhyay et al. (2017, September 7). A Practical Approach to Optimize Code Implementation [Online]. Available: https://www.einfochips.com/wp-content/uploads/resources/a-practical-approach-to-optimize-code-implementation.pdf.

[21] Michael E. Lee, “Optimization of Computer Programs in C”, Ontek Corporation, Laguna Hills, USA, 1997.

[22] Mike Jones et al. (2017, February 27). Profile application performance in Visual Studio [Online]. Available:https://docs.microsoft.com/en-us/visualstudio/profiling/beginners-guide-to-performance-profiling?view=vs-2017.

[23] Theano Petersen et al. (2018, April 11). Analyze CPU usage [Online]. Available: https://docs.microsoft.com/en-us/visualstudio/profiling/cpu-usage?view=vs-2017.

[24] Genevieve Warren et al. (2018, February 1). Analyze memory usage [Online]. Available: https://docs.microsoft.com/en-us/visualstudio/profiling/analyze-memory-usage?view=vs-2017.

[25] Mark McGee et al. (2017, February 27). Beginners guide to CPU sampling [Online]. Available: https://docs.microsoft.com/en-us/visualstudio/profiling/beginners-guide-to-cpu-sampling?view=vs-2017.

[26] Nikhil Joglekar (2016, April 28). How CPU Sampling Works [Online]. Available: https://blogs.msdn.microsoft.com/devops/2016/04/28/how-cpu-sampling-works/.