International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-8 Issue-6, March 2020

Environmental Viability as a Service (Evaas) – An Eco-Friendly Cloud Service for Optimum use of Resources at Low Energy Consumption with Less Carbon Emission

Himadri Biswas, Debabrata Sarddar, Uddyalok Chakraborty

Abstract: ***Due to enormous advancements in computer networks, people choose network-based computing rather than in-house computing. Cloud computing is one of the emerging technologies that is commonly accepted and very common in the industry. Cloud infrastructure's rising demand has significantly increased data center energy consumption. The growth of data centers will result not only in the consumption of a large amount of energy but also in the crucial effect on environmental sustainability in terms of CO2 emissions. Latest studies have shown that the electricity costs used by IT departments can be up to 50 per cent of an organization's overall energy costs. Reducing energy consumption has become a common topic of research in various fields in recent years. Environmental concerns such as greenhouse gas emissions, global warming and energy prices are significant factors that have driven many researchers to carry out computing-conscious work in the area of green and energy. Except for carbon emissions, Billing[1,2] is another exciting aspect that is closely linked to energy usage, since higher or lower billing depends on energy consumption somehow-as we know cloud providers allow cloud users to access services as pay-per-use, so these services need to be optimally chosen to process the user request to optimize customer satisfaction in the distributed virtue. There may be an inequity between the consumers' real energy usage and the providers' billing records, and any false claims that can be made by each other for unfair compensation.***

Considering all the above-mentioned issues, we are proposing a Sustainable Cloud Architecture "Environmental Viability as a Service (EVaaS)"-An Eco-friendly Cloud infrastructure for optimum usage of resources at minimum energy consumption with less carbon pollution, which in effect helps to allocate cloud users under respective Eco-friendly Service Providers. Here, the EVaaS cloud infrastructure is operated solely by the Environmentally Viable Service Provider (EVSP) without any consumer and service provider involvement. All must therefore be safeguarded against any false allegation which may be made by each other in order to seek unlawful compensation.

Keywords ***: Environmental Viability as a Service, Environmentally Viable Service Provider, Viability Demand Administrator.***

Manuscript received on March 15, 2020.
Revised Manuscript received on March 24, 2020.
Manuscript published on March 30, 2020.
* Correspondence Author

Himadri Biswas*, Department of Computer Science and Engineering, University of Kalyani, Kalyani, Nadia, W.B., India. Email: mr.himadri.biswas@gmail.com

Dr. Debabrata Sarddar, Department of Computer Science and Engineering, University of Kalyani, Kalyani, Nadia, W.B., India. Email:dsarddar1@gmail.com

Mr. Uddyalok Chakraborty, Department of Computer Science, Sharda Mandir H.S. School under Banasthali Vidyapith (Deemed University), Banasthali, Tonk, Rajasthan, India. Email: cse.uddyalok92@gmail.com

I. INTRODUCTION

Cloud Computing provides a highly scalable and cost-effective computing platform for running IT applications like High Performance Computing (HPC), Web and business applications that require ever-increasing computational resources. The advent of cloud computing has quickly modified the ownership-based computing approach model to subscription-oriented computing by offering access to flexible on-demand resources and services. According to the National Institute of Standards and Technology (NIST)[3] Cloud Computing is a paradigm that allows for omnipresent, simple on-demand network access to a common pool of configurable computer resources (e.g. networks, servers, storage, applications, and services) that can be easily distributed and released with minimal management or service provider interference. The users of the cloud can store, access, and share any amount of online information. Similarly, small and medium-sized enterprises / organizations do not think about buying, planning, running, and sustaining their own computer infrastructure. Instead, they should concentrate on improving their core competencies by leveraging a range of advantages such as low cost Cloud Storage, data center experiences, on-demand storage services, quicker and cheaper software development capabilities. In the pay-as-you-go scheme, cloud providers rent out their services. Data centers are defined as a group of physical computers, each of which has its own resources (CPUs, RAM, Network, Bandwidth, and Storage). Cloud provider can have different requirements for allocating one or more data centers in different locations. Each cloud provider is therefore responsible for updating relevant parameters such as data centers PUE, carbon pollution levels and physical machines available. The cloud broker serves as the medium between customer and provider and can assist customers across the complexities of cloud service offerings, as well as providing value-added cloud services[4]. Clouds, however, are simply data centers providing subscription-based application hosting services.

It needs a high use of energy to keep their operations running. Today a typical 1000-rack data center requires 10 Megawatt of power to run. Therefore the energy cost for a data center is a significant component of its running and capital costs. Nowadays green innovations have gained a lot of attention by growing awareness and concern on environmental issues and global warming. According to Gartner, the ICT industry generates 2% of global CO2 emissions equal to the aviation industry[5], rising at a rate of 6% per annum, and with such a rate of growth they could represent 12% of global emissions by 2020[6]. U.S. data centers consume 1.5 of total energy usage in the U.S. based on the EPA survey, and its expense is $4.5 billion[6]. With the increasing popularity of cloud computing, many consumers and businesses are utilizing and providing cloud services that have resulted in rising large-scale data centers with high operating costs, massive quantities of electricity, and substantial carbon dioxide emissions. Many research works done to significantly reduce energy consumption, but they do not ensure that CO2 emissions as a whole are minimised. Sustainable protection is a big concern. We therefore suggest a user-oriented Carbon Conscious Environmentally Viable Cloud Architecture with less physical node utilization to safely minimize the carbon footprint of cloud computing without compromising the quality of service (such as efficiency, responsiveness and availability) provided by multiple cloud providers. Our architecture is designed to help consumers in obtaining an Eco-friendly service provider with less carbon pollution at minimal energy consumption. So many research works has been done on Green Cloud computing that only provides the Green computing platform, but there is no such verification method claimed as Green. Our proposed work not only offers the Environmentally Sustainable Cloud services, but is also used to check whether Eco-friendly service providers are capable of delivering the services to support the ecosystem or not.

II. RELATED WORKS

There has been a lot of work and analysis in decreasing energy usage and green cloud computing, some of them we have briefly discussed here. Saurbh kumar Garg et al, has proposed a carbon-aware Green system that addresses the environmental problem and aims to reduce the carbon footprint generated by cloud computing, focusing on the centralized role of broker's in cloud environments[7]. Farrahi Moghaddam et al [8] proposed a model focused on resource usage and performance monitoring counter (PMC) for calculating energy use and carbon footprint. Kansal et al.[9] that measure the power of the VM. It used tracking of resource usage by a VM, such as processors, memory, disk arrays and so on, by measuring the power consumption during the time of use, calculating the active energy use of a VM. Farrahi Moghaddam et al [10] have researched VM migration in WAN and suggested a genetic algorithm-based approach.

Reducing energy consumption and carbon footprint is highly beneficial, but may lead to undesirable performance degradation. In [11] a concept was proposed that described trade-off between carbon emissions and QoS in cloud systems that are located in various geographical locations across various datacenters. To solve the optimization problem they used a sub gradient method. Saad Mustafa et al [12] studied in Best Fit Decreasing for VM placement and migration on the assessment of energy quality. In 2013, Nquyen Quang-Hung et al proposed a genetic algorithm for power-aware resource allocation (GAPA) to solve the static virtual machine allocation problem [13]. Peter Xiang Geo et al[14] proposed FORTE (Flow Optimization based application routing and traffic engineering framework) to monitor the three-way trade between average working time, energy costs and carbon emissions. Fanxin Kong and Liu are studying the question of green-carbon-aware power management in modern data centers incorporating intermittent or green energy sources into their power supply. Here, the Green-energy-aware work is divided in to four categories, green-energy-aware scheduling of workloads, green-energy-conscious virtual machine management, green-energy-conscious resource planning and interdisciplinary [15]. In 2011, Ryan Jansen et al [16] analyzed the effect of various scheduling policies on energy consumption in cloud computing and explored seven scheduling techniques Round-Robin, Stripping, Packing, Count-based Load Balancing, Ratio-based Load Balancing, Watts per Core and Cost per Core, based on their simulations, "Stripping" had the higher energy consumption and the lower the "watts per Core" [16]. Yongqiang Wu et al proposed an algorithm in 2012, based on simulated VM placement annealing [17]. Two TUE and ITUE metrics for measuring the datacenter's energy efficiency proposed here [18]. Two other metrics proposed by authors in [19] are DWPE and PUE. In [20], a two-phase carbon-conscious cloud broker with different carbon emission factors has been proposed in distributed regional datacenters that seek to minimize energy and carbon emissions by considering data center efficiency and power output. Here are many metrics for measuring the greening and efficiency of electricity and energy use, some of which are: PUE (Power Use Effectiveness)[18], DCiE (Data Center Infrastructure Efficiency) [21], CUE (Carbon Use Effectiveness) [18], WUE (Water Usage Effectiveness) [22], DCPI (Data Center Productivity Index), ERE (Energy Reuse Effectiveness)[23],

ERF (Energy Reuse Factor), GEC(Green Energy Coefficient) [24], TUE (Total Power Usage Effectiveness) [18], EUE (Energy Usage Efficiency) [25], DWPE (Data Center Workload Power Efficiency) [19], DCEP (Data Center Energy Productivity) [26], CPE (Compute Power efficiency) [27]. In [28] a genetic algorithm called Hybrid Energy Aware algorithm (HEA) used for VM allocation is used to achieve green cloud computing by considering the QoS (SLA and response time). In [29] VM placement problem is formulated as a restricted multi-objective optimization problem with the goal of optimizing income, reducing load balancing and resource waste.

A proposed ant colony optimization algorithm for VM placement with the goal of minimizing the power consumption and resource wastage assumed by heterogeneous VMs and PMs presented here [30]. Bharti Wadhwa et al [31] suggested a strategy for minimizing carbon footprint and energy consumption called Carbon Efficient VM Placement and Migration Method (CEPM) by considering dispersed data centers and variations in carbon emission levels and energy sources. Sura Khalil Abd et al [32] proposed a DNA-based Fuzzy Genetic Algorithm for work scheduling to reduce power consumption and optimize the usage of resources. A genetic algorithm which the authors presented in [33] called ETSG for task scheduling to reduce energy consumption. Consolidation of workloads and consolidation of complex virtual machines addressed respectively in [34] & [35]. Another energy saving service provided by Himadri Biswas et al. [36] for Maximum Resource Usage at Minimum Energy Consumption Without Live Migration of Virtual Machines using the Virtual Server Principle. So, a lot of research work has been done on Green Cloud computing which only offers the Green computing platform, but there is no such verification mechanism as Green believed. Our proposed work not only offers the Environmentally Sustainable Cloud services, but is also used to check the Eco-friendly service providers whether capable of delivering the services to support the ecosystem or not.

Safety as a Service (SFaaS) [37], Verification as a Service (VaaS) [1] and Power Management as a Service Model [36] are proposed where Security Service Service Provider (SSP), Verification Service Provider (VSP) and Power Management Service Provider (PMSP) are likely to function as a TTP except that users and CSPs can not communicate directly with each other without the permission of the respective SSP, VSP and PMSP. So, Environmental Viability as a service (EVaaS) is the missing link for monitoring the power effectiveness at low carbon emission where EVSP not only offers the Environmentally Sustainable Cloud services, but acts as a mediator as well as a verifier between users and CSPs for delivering the services to support the ecosystem.

III. OVERVIEW AND RATIONALE

A. Cloud Service Models

Cloud computing technology models are-

Software as a Service (SaaS): This SaaS model offers a ready-made application along with the necessary software, operating system, hardware and network.

Platform as a Service (PaaS): This PaaS model offers hardware, network, and operating system, while the user can install or create their own applications and software's.

Infrastructure as a Service (IaaS): Also this IaaS model includes the hardware and networks, the user designs or constructs their own operating systems, software and applications.

B. Proposed Service Model

Fig. 1.Cloud Service Models

Environmental Viability as a Service (EVaaS)

EVaaS under Private Cloud provides an Environmental-friendly and Quality of Cloud services to the cloud users for optimum use of resources at low energy consumption with less Carbon Emission. The main concept is to verify the required resources provided by the Viable Cloud service Providers (CSP) to their users as per service level agreements (SLA) without any partiality either or both sides and no chance of intervention between the users and the Service Providers directly, because once a user or a service provider registered in EVaaS, must go through this service.

C. Environmentally Viable Service Provider (EVSP):

EVSP under EVaaS, plays the important role of assigning the Users under their Eco-Friendly Cloud Service Providers. Before assigning the users to their CSPs', EVSP collect the required resources from their respective CSPs' for evaluating whether they are capable or not to provide an environmental sustainable cloud services.

D. Viability Demand Administrator (VDA):

Upon receiving the periodic signals or available resources from the CSPs', VDA create, manage and store the Viability demand CSPs' into Viability Demand Cloud Services (VDCS) module based on the analysis of received records through Power Efficiency Monitor & Carbon Emission Monitor.

IV. PROPOSED WORK

Emerging cloud computing infrastructures provide pay-based computing resources as needed. Maximum resource utilization using less power consumption with less Carbon emission is therefore one of the most challenging features to improve cloud usability.

Figure 2 describes our proposed service model "**EVaaS**", where EVSP helps to assign the VMs to PMs in such a way that resource utilization should be maximum at less energy consumption, as a result carbon emission should be minimum. As we know that there are different cloud service models and their intended service providers available in the market, but most of them are not aware about environmental sustainability due to huge amount of carbon emission from their datacenters. Our aim in this work is to provide a service model EVaaS which is viable for the environment and where the users if they want must be going through this model before getting any services from the CSP's and in that case no CSPs should provide any final intimation to the users without getting approval from EVSP under EVaaS.

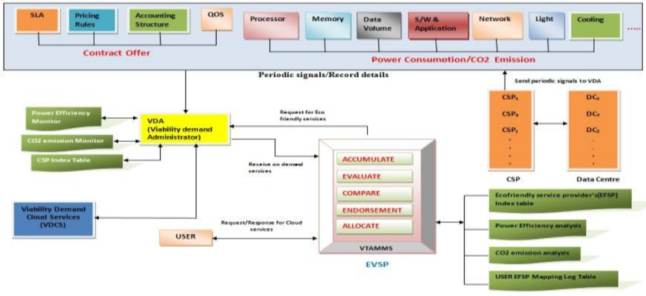


Fig. 2.Proposed Architecture of Environmetal Viability as a Service (EVaaS) Model

A. Working procedure of EVaaS Model

Figure 2 represents the working model of EVaaS, where the CSPs' send the periodic signals or record details to VDA demanding as Viable Service Providers. Upon receiving the records from CSPs' VDA evaluates the records by its Power Efficiency & Carbon Emission Monitor for checking the low power consumption and less Carbon emission by the processing node(s) and based on the evaluation VDA decides whether the CSP is viable or not. If viable then store the record details into the VDCS module and maintains an index table of viable CSPs'. On the other hand, after receiving the on demand service requests from VDA, EVSP maintains an index table of Eco-friendly Service Providers (EFSP).When a user requests for a cloud service to the EVSP, it checks the existence of its CSP from its EFSPs' index table and if exists EVSP forwards the request to VDA for the dedicated CSP to provide eco-friendly services. Now the VTAMMS (Viability Test, Approval and Mapping Management System) collect the required resource details from VDA by its ACCUMULATE module. Now the EVALUATE module of VTAMMS compute the power consumption and carbon emission about the resources related to the corresponding CSPs'. Based on the evaluated results VTAMMS verify the capability of the CSP for viable services. If capable for optimum use of resources at low energy consumption with less Carbon Emission, then approved as Eco-friendly service provider and sends an acknowledgement to the CSP through VDA for assigning its requested users to provide the cloud services.

V. ALGORITHM, FLOWCHART, PSEUDO CODE

A. Algorithm: Power Consumption & CO2 Emission calculation

```
1. User requests for a cloud service through EVSP
2. EVSP checks the existence of its CSP from its
   Eco-friendly Service Provider's list.
3. If the CSP exists –
   3.1 EVSP forwards request to VDA
   3.2 VDA forwards the request to dedicated CSP's
       for Eco-friendly services.
   3.3 CSP check for user authentication & SLA.
   3.4 If authentic –
       3.4.1 Accepted for services and send an
             acknowledgement to EVSP through VDA.
       3.4.2 CSP sends the periodic signal to VDA for
             offering Environmentally viable services.
       3.4.3 VDA checks the Power efficiency and
             $Co_2$ Emission rate.
       3.4.4 If within threshold --
             3.4.4.1 VDA Store the records into VDCS
                     offered as Viability Demand
                     services.
       3.4.5 Else –
             3.4.5.1 Go to Step 3.4.2.
       3.4.6 VTAMMS of EVSP collect the record
             details (which sent by CSP
             periodically to VDA for offering
             Eco-friendly cloud services) from VDCS
             through VDA.
       3.4.7 VTAMMS evaluates the Power efficiency and
             $CO_2$ emission analysis based on the data
             collected from VDCS.
       3.4.8 Based on the analysis VTAMMS compare with
             the received record Details (sent by the
             dedicated CSP), whether it is capable or not to
             provide Eco-friendly Cloud services.
       3.4.9 If capable with minimum CO2 emission at low
             Power Consumption –
             3.4.9.1 With the concern of EVSP, VTAMMS
                     approve the CSP as EFSP.
             3.4.9.2 EVSP sends an acknowledgement to the
                     CSP for assigning the requested user.
             3.4.9.3 Mapping is done between the User and
                     the EFSP
       3.4.10 Else –
             3.4.10.1 Go to Step 3.1
   3.5 Else –
       3.5.1 "User not accepted for
             service"— message returns to the user


       3.5.2 Go to Step 1.
4. Else –
   4.1 "No such CSP exists"—message returns to the user.
   4.2 Go to Step 1.
5. End.
```

B. Flowchart :

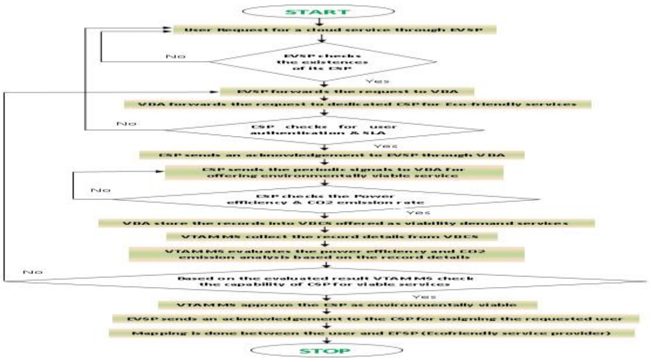


Fig. 3.Proposed Flowchart of Environmetal Viability as a Service (EVaaS) Model

C. Pseudo code: Compute_Power Consumption & CO2 Emission

Notations:

AVG_{wl}- Average workload of an instance, **P_{speed}**– Processor speed of an instance (Amount of data processing per sec.), **N_TH_{value}** –Threshold value of a node instance, **T**– Set of Tasks, **P** – Set of Processes, **$NODE_{ins}$** – Set of node instances required for process execution, **E_{CO2}** – CO_2 emission per sec. **P_{consmp}** –Power consumption per sec., **p** –Number of processes, **n** – Number of instances, **k** – Number of tasks of a process, **l** – Number of communications between two tasks

t – Average time for each inter task communications

m – Number of communications between two processes

r – Average time for each inter process communications

q – Number of communications between two nodes

s – Average time for each inter node communications

```
Begin GET_NODE_INSTANCE
Data:
      $AVG_{wl}$, $P_{speed}$, $N\_TH_{value}$, $NODE_{ins}$ initialize to zero
While true do
   For each $NODE_{ins}$ [i = 1 to n] do
      $AVG_{wl} \leftarrow$ Avg_Work_Load ($NODE_{ins}$ [i])
      $P_{speed} \leftarrow$ Processor_Speed ($NODE_{ins}$ [i])
      $N\_TH_{value} \leftarrow AVG_{wl} * P_{speed}$
      Store ($NODE_{ins}$ [i], $N\_TH_{value}$)
      i$\leftarrow$i+1
   End for
End while
End GET_NODE_INSTANCE
Begin PROCESS_INSTANCE
Data:  NODE, STORAGE, MEMORY, NETWORK,
       $IT_{resources}$, $OTHER_{resources}$, $REQUIRED_{resources}$,
       $POWER_{consmp}$, $EMISSION_{CO2}$ Initialize to zero
While true do
   For each instance [i=1 to n] do
      If ($NODE_{ins}$[i]<= $N\_TH_{value}$)
         For each  process[j=1 to p] do
            For each task [t=1 to k]
               NODE $\leftarrow$CPUreq(T[t])
               STORAGE $\leftarrow$ $STORAGE_{req}$(T[t])
               MEMORY $\leftarrow$ MEMORY $_{req}$(T[t])
               NETWORK $\leftarrow$$BANDWIDTH_{req}$(T[t])
               $IT_{resources}$ $\leftarrow$$SW\_APPLN_{req}$(T[t])
               $OTHER_{resources}$$\leftarrow$$LIGHT\_COOLING_{req}$(T[t])
               $REQUIRED_{resources}$$\leftarrow$$RESOURCE_{req}$(NODE,
               STORAGE, MEMORY, NETWORK,
               $IT_{resources}$, $OTHER_{resources}$)
             Store($NODE_{ins}$[i]P[j]T[t], $REQUIRED_{resources}$)
             $POWER_{consmp}$ $\leftarrow$P0WER$_{req}$($NODE_{ins}$[i]P[j]T[t])
            $EMISSION_{CO2}$$\leftarrow$$CO2_{Emission}$($NODE_{ins}$[i]P[j]T[t])
             t$\leftarrow$t+1
             end for
           j$\leftarrow$j+1
           end for
      endif
   i$\leftarrow$i+1
   end for
End while
End PROCESS_INSTANCE
// For PROCESS_INSTANCE
Begin $CPU_{req}$
Data: Tins -Task instance, $SET_{CPU}$ - Setup CPU requirement
$SET_{CPU}$$\leftarrow$VDA.$GET_{CPUreq}$[$T_{ins}$]
Return $SET_{CPU}$
End $CPU_{req}$
Begin $STORAGE_{req}$
Data:$T_{ins}$-Task instance, $SET_{storage}$- Setup Storage requirement
$SET_{storage}$$\leftarrow$VDA.$GET_{STORAGEreq}$[Tins]
Return $SET_{storage}$
End $STORAGE_{req}$
Begin $MEMORY_{req}$
Data:   $T_{ins}$ -Task instance,
        $SET_{memory}$ - Setup Memory requirement
$SET_{memory}$$\leftarrow$VDA.$GET_{MEMORYreq}$[$T_{ins}$]
Return $SET_{memory}$
End $MEMORY_{req}$
Begin $BANDWIDTH_{req}$
Data:   Tins -Task instance,
        $SET_{bandwidth}$ -Setup Bandwidth requirement
$SET_{memory}$$\leftarrow$VDA.$GET_{BANDWIDTHreq}$[$T_{ins}$]
Return $SET_{bandwidth}$
End $BANDWIDTH_{req}$
Begin $SW\_APPLN_{req}$
Data:   $T_{ins}$ -Task instance, $SET_{SW\_APPLN}$-Setup software &
        Applications requirement
$SET_{SW\_APPLN}$ $\leftarrow$VDA.$GET_{SW\_APPLNreq}$[$T_{ins}$]
Return $SET_{SW\_APPLN}$
End $SW\_APPLN_{req}$
Begin $LIGHT\_COOLING_{req}$
Data:   $T_{ins}$ -Task instance, $SET_{Light\_Cooling}$ -Setup light &
        cooling requirement
$SET_{Light\_Cooling}$$\leftarrow$VDA.$GET_{LIGHT\_COOLINGreq}$[$T_{ins}$]
Return $SET_{Light\_Cooling}$
End $LIGHT\_COOLING_{req}$
Begin $RESOURCE_{req}$
Data: $SET_{Required\_Resource}$ -Setup Resource requirement
$SET_{Required\_Resource}$$\leftarrow$VDA.$GET_{RESOURCEreq}$ ($SET_{CPU}$, $SET_{storage}$,
$SET_{memory}$, $SET_{bandwidth}$, $SET_{SW\_APPLN,}$ $SET_{Light\_Cooling}$)
Return $SET_{Required\_Resource}$
End $RESOURCE_{req}$
Begin $POWER_{req}$
Data:   $T_{ins}$ -Task instance, $P_{ins}$ -Process instance, $NODE_{ins}$
        -Node instance, $SET_{power}$ -Setup Power Requirement
$SET_{power}$$\leftarrow$VDA.$GET_{POWEREreq}$ ($NODE_{ins}$, $P_{ins}$, $T_{ins}$)
Return $SET_{power}$
End $POWER_{req}$
Begin $CO2_{Emission}$
Data: $T_{ins}$-Task instance, $P_{ins}$-Process instance, $NODE_{ins}$
      -Node instance, $SET_{CO2Emission}$ -Setup CO2 Emission
$SET_{CO2Emission}$$\leftarrow$VDA.$GET_{EMITTEDco2}$ ($NODE_{ins}$, $P_{ins}$, $T_{ins}$)
Return $SET_{CO2Emission}$
End $CO2_{Emission}$
//For GET_NODE_INSTANCE
Begin Avg_Work_Load
Data: $INS_{id}$ –Instance Id, AWL – Average Work Load
AWL$\leftarrow$VDA.$GET_{workload}$($INS_{id}$)
Return AWL
End Avg_Work_Load
Begin Processor_Speed
Data: $INS_{id}$ – Instance Id, PS – Processor speed
PS$\leftarrow$VDA.$GET_{speed}$($INS_{id}$)
Return PS
End Processor_Speed
```

Begin **EVALUATE_Power consumption_CO2 Emission**
Data**:**

$\mathbf{R_{consmp}}$ –Consumed resources from VDA, $\mathbf{Task_{Pconsmp}}$- Power consumption of a task, $\mathbf{P_{TaskPconsmp}}$ –Power consumption of all tasks of a process initialized to zero, $\mathbf{Task_{Eco2}}$ –Co2 emission from a task, $\mathbf{P_{TaskEco2}}$ –Co2 emission of all tasks of a process initialized to zero, $\mathbf{P_{consmp}ITC_{ab}}$ –Power consumption due to inter task communication between tasks a & b, $\mathbf{P_{consmp}ITC}$ – Power consumption due to inter task communication initialized to zero, $\mathbf{E_{CO2}ITC_{ab}}$ -CO2 emission due to inter task communication between tasks a & b, $\mathbf{E_{CO2}ITC}$– Co2 emission due to inter task communication initialized to zero, $\mathbf{P_{consmp}IPC_{cd}}$–Power consumption due to inter process communication between Processes c & d, $\mathbf{P_{consmp}IPC}$ – Power consumption due to inter process communication initialized to zero, $\mathbf{E_{CO2}IPC_{cd}}$ -CO2 emission due to inter process communication between processes c & d, $\mathbf{E_{CO2}IPC}$ -CO2 emission due to their inter process communication initialized to zero, $\mathbf{P_{consmp}INC_{xy}}$ – Power consumption due to inter node communication between nodes x & y, $\mathbf{P_{consmp}INC}$ – Power consumption due to inter node communication initialized to zero, $\mathbf{E_{CO2}INC_{xy}}$ -CO2 emission due to inter node communication between nodes x & y, $\mathbf{E_{CO2}INC}$ -CO2 emission due to their inter node communication initialized to zero, $\mathbf{P_{Pconsmp}}$ – Power consumption of a process initialized to zero, $\mathbf{P_{ECO2}}$ – CO2 emission of a process initialized to zero
$\mathbf{PSU_{Pconsmp}}$ – Power consumption due to execution of all processes submitted by the user, $\mathbf{PSU_{ECO2}}$– Co2 emission due to execution of all processes submitted by the user, $\mathbf{P_{consmp}NODE_{ins}}$ – Power consumption due to execution of all processes on a single node, $\mathbf{E_{CO2}NODE_{ins}}$ –CO2 emission due to execution of all processes on a single node, $\mathbf{PSU_{MN\_Pconsmp}}$ - Power consumption due to execution of all processes on multi node, $\mathbf{PSU_{MN\_ECO2}}$ - CO2 emission due to execution of all processes on multi node

```
For each instance [i=1 to n] do
  For each  process[j=1 to p] do
        For each task [t=1 to k]
          $R_{consmp} \leftarrow GET_{Resource}(Resource)$
          Store $(NODE_{ins}[i]P[j]T[t], R_{consmp})$
          $Task_{Pconsmp} \leftarrow GET_{Power}(R_{consmp})$
          $Task_{Eco2} \leftarrow GET_{ECO2}\ (R_{consmp})$
          $P_{TaskPconsmp} \leftarrow P_{TaskPconsmp} +$
                     $COMPUTE_{Power}(Task_{Pconsmp})$
          $P_{TaskEco2} \leftarrow P_{TaskEco2} +$
                   $COMPUTE_{CO2Emission}(Task_{Eco2})$
        t←t+1
        End for
      For each task[a=1 to k-1] do
          For each task [b=a+1 to k] do
            If (a interact with b)
              $P_{consmp}ITC_{ab} \leftarrow (l*t)*P_{consmp}$
              $P_{consmp}ITC \leftarrow P_{consmp}ITC + P_{consmp}ITC_{ab}$
            Else
              $P_{consmp}ITC \leftarrow P_{consmp}ITC$
            End if

            If (a interact with b)
              $E_{CO2}ITC_{ab} \leftarrow (l*t)*E_{CO2}$
              $E_{CO2}ITC \leftarrow E_{CO2}ITC + E_{CO2}ITC_{ab}$
            Else
              $E_{CO2}ITC \leftarrow E_{CO2}ITC$
            End if
          b←b+1
          End for
      a←a+1
      End for
      $P_{Pconsmp}[j[ \leftarrow P_{TaskPconsmp} + P_{consmp}ITC$
      $P_{ECO2}[j] \leftarrow P_{TaskEco2} + E_{CO2}ITC$
      $PSU_{Pconsmp} \leftarrow PSU_{Pconsmp} + P_{Pconsmp}[j]$
      $PSU_{ECO2} \leftarrow PSU_{ECO2} + P_{ECO2}[j]$
  j←j+1
  End for

       For each process[c=1 to p-1] do
        For each process [d=c+1 to p] do
          If (c interact with d)
            $P_{consmp}IPC_{ab} \leftarrow (m*r)*P_{consmp}$
            $P_{consmp}IPC \leftarrow P_{consmp}IPC + P_{consmp}IPC_{ab}$
          Else
            $P_{consmp}IPC \leftarrow P_{consmp}IPC$
          End if
          If (c interacts with d)
            $E_{CO2}IPC_{ab} \leftarrow (m*r)*E_{CO2}$
            $E_{CO2}IPC \leftarrow E_{CO2}IPC + E_{CO2}IPC_{ab}$
          Else
            $E_{CO2}IPC \leftarrow E_{CO2}IPC$
          End if
        d←d+1
        End for
      c←c+1
      End for
  $PSU_{Pconsmp} \leftarrow PSU_{Pconsmp} + P_{consmp}IPC$
  // Power consumption due to execution of all processes on a single node
  $P_{consmp}NODE_{ins}[i] \leftarrow PSU_{Pconsmp}$
  $PSU_{ECO2} \leftarrow PSU_{ECO2} + E_{CO2}IPC$
  // $CO_2$ emission due to execution of all processes on a single node
  $E_{CO2}NODE_{ins}[i] \leftarrow PSU_{ECO2}$
  i←i+1
  End for
For each node[x=1 to n-1] do
  For each node [y=x+1 to n] do
        If (x interact with y)
          $P_{consmp}INC_{xy} \leftarrow (q*s)*P_{consmp}$
          $P_{consmp}INC \leftarrow P_{consmp}INC + P_{consmp}INC_{xy}$
        Else  $P_{consmp}INC \leftarrow P_{consmp}INC$
        End if
        If (x interacts with y)
          $E_{CO2}INC_{xy} \leftarrow (q*s)*E_{CO2}$
          $E_{CO2}INC \leftarrow E_{CO2}INC + E_{CO2}INC_{xy}$
        Else
          $E_{CO2}INC \leftarrow E_{CO2}INC$
        End if
        y←y+1
        End for
    x←x+1
    End for
For each $NODE_{ins}[i=1\ to\ n]$ do
  $PSU_{MN\_Pconsmp} \leftarrow PSU_{MN\_Pconsmp} + P_{consmp}NODE_{ins}[i]$
  $PSU_{MN\_ECO2} \leftarrow PSU_{MN\_ECO2} + E_{CO2}NODE_{ins}[i]$
i←i+1
End for
```

```
// Power consumption due to execution of all processes on multi node
$PSU_{MN\_Pconsmp} \leftarrow PSU_{MN\_Pconsmp} + P_{consmp}INC$
// $CO_2$ emission due to execution of all processes on multi node
$PSU_{MN\_ECO2} \leftarrow PSU_{MN\_ECO2} + E_{CO2}INC$
End EVALUATE_Power consumption_CO2 Emission
// For EVALUATE_Power consumption_CO2 Emission
Begin $GET_{Resource}$
Data: Resource, Consmp
Resource $\leftarrow$ VDA.GETResource_Req($SET_{CPU}$, $SET_{storage}$, $SET_{memory}$, $SET_{bandwidth}$, $SET_{SW\_APPLN}$, $SET_{Light\_Cooling}$)
Consmp $\leftarrow$ EVSP.VTAMMS.ACCUMULATE(Resource)
Return Consmp
End $GET_{Resource}$
Begin $GET_{Power}$
Data: $R_{consmp}$, Consmp
$R_{consmp} \leftarrow$ VDA.GETPower_Req[$NODE^{ins}$[i]P[j]T[t]]
Consmp $\leftarrow$ EVSP.VTAMMS.ACCUMULATE($R_{consmp}$)
Return Consmp
End $GET_{Power}$
Begin $GET_{ECO2}$
Data: $R_{consmp}$, Consmp
$R_{consmp} \leftarrow$ VDA.$GET_{Emitted\_CO2}$[$NODE_{ins}$[i]P[j]T[t]]
Consmp $\leftarrow$ EVSP.VTAMMS.ACCUMULATE($R_{consmp}$)
Return Consmp
End $GET_{ECO2}$
Begin $COMPUTE_{Power}$
Data: $Task_{Pconsmp}$, Consmp, $P_{Speed}$, $P_{consmp}$, $O_{Pconsmp}$ – Power
    consumption in second from other equipments by each task
$Task_{Pconsmp} \leftarrow [\ Task_{Pconsmp} / P_{Speed}) * P_{consmp}] + O_{Pconsmp}$
Consmp $\leftarrow$ EVSP.VTAMMS.EVALUATE($Task_{Pconsmp}$)
Return Consmp
End $COMPUTE_{Power}$
Begin $COMPUTE_{CO2Emission}$
Data: $Task_{ECO2}$, Consmp, $P_{Speed}$, $E_{CO2}$, $O_{ECO2}$ – $CO_2$ emission in
    second from other equipments by each task
$Task_{ECO2} \leftarrow [\ Task_{ECO2} / P_{Speed}) * E_{co2}] + O_{ECO2}$
Consmp $\leftarrow$ EVSP.VTAMMS.EVALUATE($Task_{ECO2}$)
Return Consmp
End $COMPUTE_{CO2Emission}$
```

VI. DETAILED ANALYSIS OF EVAAS MODEL

A. Computation of Total Power Consumption & CO2 Emission by VTAMMS Module:

User submits a number of processes and each process contains a number of tasks. Each process executes under the same node (processor) or different nodes. The process is assigned to the node at low power consumption and least CO_2 emission based on the threshold value of that node, threshold value of a node is calculated as a product of the average workload of all the nodes and the processor speed, which is constant and depends on the processing capability of that node relative to the processing capability of all other nodes. So, before calculating the overall power consumption and CO_2 emission of all processes we have to calculate the power consumption and CO_2 emission due to inter task communications, inter process communications as well as inter node communications.

***Power Consumption due to the Inter Task Communication* (P_{consmp}ITC):** During the execution of the process, if two tasks communicates l times and if the average times for each inter task communication is t, then P_{consmp}ITC for the two tasks is (l × t) × P_{consmp}, where P_{consmp} is the Power Consumption per second.

***CO_2 emission due to the Inter Task Communication* (E_{CO2}ITC):** During the execution of the process, if two tasks communicates l times and if the average times for each inter task communication is t, then E_{CO2}ITC for the two tasks is (l × t) × E_{CO2}, where E_{CO2} is the CO_2 emission per second.

***Power Consumption due to the Inter Process Communication* (P_{consmp}IPC):** During the execution of the processes, if two processes communicates m times and if the average times for each inter process communication is r, then P_{consmp}IPC for the two processes is (m × r) × P_{consmp}, where P_{consmp} is the Power Consumption per second.

***CO_2 emission due to the Inter Process Communication* (E_{CO2}IPC):** During the execution of the processes, if two processes communicates m times and if the average times for each inter process communication is r, then E_{CO2}IPC for the two processes is (m × r) × E_{CO2}, where E_{CO2} is the CO_2 emission per second.

***Power Consumption due to the Inter Node Communication* (P_{consmp}INC):** During the execution of the processes, if two processing nodes communicates q times and if the average times for each inter node communication is s, then P_{consmp}INC for the two nodes is (q × s) × P_{consmp}, where P_{consmp} is the Power Consumption per second.

***CO_2 emission due to the Inter Node Communication* (E_{CO2}INC):** During the execution of the processes, if two processing nodes communicates q times and if the average times for each inter node communication is s, then E_{CO2}INC for the two processes is (q × s) × E_{CO2}, where E_{CO2} is the CO_2 emission per second.

Now, we can calculate the ***Total Power Consumption*** **($T_{PConsmp}$)** of all the processes submitted by the user. i.e,

$$T_{PConsmp} = (p \times P_{Pconsmp}) + P_{consmp}IPC + P_{consmp}INC,$$

Where, **p** is the number of processes submitted by the user, **$P_{Pconsmp}$** is the Power Consumption of a process execution, **P_{consmp}IPC** is the Power consumption due to inter process communications, **P_{consmp}INC** is the Power consumption due to inter node communications.

Similarly, we can find out the ***Total CO_2 emission*** **(T_{ECO2})** of all processes submitted by the user, i.e,

$$T_{ECO2} = (p \times P_{ECO2}) + E_{CO2}IPC + E_{CO2}INC,$$

Where, **p** is the number of processes submitted by the user, **P_{ECO2}** is the CO_2 emission of a process execution, **E_{CO2}IPC** is the CO_2 emission due to inter process communications, **E_{CO2}INC** is the CO_2 emission due to inter node communications.

Let us know that, each process contains a number of tasks. So, the ***Power consumption of processing each task*** of individual processes on their assigned node(s),

$$Task_{Pconsmp} = (TC_{bytes} / P_{Speed}) \times P_{consmp} + O_{Pconsmp} \text{ ------- } (1.1)$$

Where, **$Task_{Pconsmp}$** – Power consumption of processing each task, **TC_{bytes}** -Required amount of computation (Number of instructions executed + CPU Usage + memory usage + S/w and applications Usage + network usage + data storage usage) by each task in bytes, **P_{Speed}** – Speed of the processor (Amount of data in bytes that the node can process/sec)

P_{consmp} - Power consumption per second, **$O_{Pconsmp}$** – Power consumption from other equipments by each task

Similarly, we can find out the ***CO_2 emission of processing each task*** of individual processes on their assigned node(s),

$$\mathbf{Task_{Eco2} = (TC_{bytes} / P_{Speed})\ x\ E_{CO2} + O_{ECO2}} \text{ ------------ } \mathbf{(1.2)}$$

Where, E_{CO2} – CO_2 emission per second, O_{ECO2} - CO_2 emission from other equipments by each task

If k is the number of tasks of a process, a & b are the pair of tasks for inter task communication, then,

Power Consumption due to the Inter Task Communications of a process:

$$P_{consmp}ITC = \sum_{a=1}^{k-1} \sum_{b=a+1}^{k} P_{consmp}ITC_{ab},\ P_{consmp}ITC_{ab} \begin{cases} 1 & a \rightarrow b \\ 0 & a \nrightarrow b \end{cases} \text{-- } \mathbf{(2.1)}$$

CO_2 emission due to the Inter Task Communications of a process:

$$E_{CO2}ITC = \sum_{a=1}^{k-1} \sum_{b=a+1}^{k} E_{CO2}ITC_{ab},\ E_{CO2}ITC_{ab} \begin{cases} 1 & a \rightarrow b \\ 0 & a \nrightarrow b \end{cases} \text{ ---- } \mathbf{(2,2)}$$

From (1.1) and (2.1), ***the total Power Consumption due to execution of all tasks of a Process,***

$$P_{Pconsmp} = (k\ x\ Task_{Pconsmp}) + P_{consmp}ITC \qquad \text{---------- } \mathbf{(3.1)}$$

From (1.2) and (2.2), ***the total CO_2 emission due to execution of all tasks of a Process,***

$$P_{ECO2} = (k\ x\ Task_{Eco2}) + E_{CO2}ITC \qquad \text{---------- } \mathbf{(3.2)}$$

If p is the number of processes submitted by the user, c & d are the pair of processes for inter process communication, then,

Power Consumption due to the Inter Process Communications:

$$P_{consmp}IPC = \sum_{c=1}^{p-1} \sum_{d=c+1}^{p} P_{consmp}IPC_{cd},\ P_{consmp}IPC_{cd} \begin{cases} 1 & c \rightarrow d \\ 0 & c \nrightarrow d \end{cases} \text{ ---}\mathbf{(4.1)}$$

CO_2 emission due to the Inter Process Communication:

$$E_{CO2}IPC = \sum_{c=1}^{p-1} \sum_{d=c+1}^{p} E_{CO2}IPC_{cd},\ E_{CO2}IPC_{cd} \begin{cases} 1 & c \rightarrow d \\ 0 & c \nrightarrow d \end{cases} \text{ ----}\mathbf{(4.2)}$$

From (3.1) and (4.1), the ***Power Consumption due to execution of all Processes*** submitted by the user,

$$PSU_{Pconsmp} = (p\ x\ P_{Pconsmp}) + P_{consmp}IPC,\ p>=1,\ P_{consmp}IPC >=0 \text{ ------------ } \mathbf{(5.1)}$$

From (3.2) and (4.2), the ***CO_2 emission due to execution of all Processes*** submitted by the user,

$$PSU_{ECO2} = (p\ x\ P_{ECO2}) + E_{CO2}IPC,\ p>=1,\ E_{CO2}IPC >=0 \text{ ---}\mathbf{(5.2)}$$

So, from (5.1) it is clear that the ***Power Consumption due to execution of all Processes submitted by the user on a single node***, $P_{consmp}NODE_{ins} = PSU_{Pconsmp}$ ------------**(6.1)**

Similarly, from (5.2) the ***CO_2 emission due to execution of all Processes submitted by the user on a single node,*** $E_{CO2}NODE_{ins} = PSU_{ECO2}$ ------------**(6.2)**

If n is the number of nodes processed for execution of all the processes submitted by the user; x & y are the pair of nodes for inter node communication, then,

Power Consumption due to the Inter Node Communications:

$$P_{consmp}INC = \sum_{x=1}^{n-1} \sum_{y=x+1}^{n} P_{consmp}INC_{xy},\ P_{consmp}INC_{xy} \begin{cases} 1 & x \rightarrow y \\ 0 & x \nrightarrow y \end{cases} \text{ --}\mathbf{(7.1)}$$

CO_2 emission due to the Inter Node Communications:

$$E_{CO2}INC = \sum_{x=1}^{n-1} \sum_{y=x+1}^{n} E_{CO2}INC_{xy},\ E_{CO2}INC_{xy} \begin{cases} 1 & x \rightarrow y \\ 0 & x \nrightarrow y \end{cases} \text{ --- } \mathbf{(7.2)}$$

So, ***Total Power consumption due to execution of all processes submitted by the user on multi node***,

$$\begin{aligned} PSU_{MN\_Pconsmp} &= (n\ x\ P_{consmp}NODE_{ins}) + P_{consmp}INC \\ &= (n\ x\ PSU_{Pconsmp}) + P_{consmp}INC, \\ &\quad n >= 1,\ P_{consmp}INC>=0 \text{ -----------}\mathbf{(8.1)} \end{aligned}$$

Total CO_2 emission due to execution of all processes submitted by the user on multi node,

$$\begin{aligned} PSU_{MN\_ECO2} &= (n\ x\ E_{CO2}NODE_{ins}) + E_{CO2}INC \\ &= (n\ x\ PSU_{ECO2}) + E_{CO2}INC, \\ &\quad n >= 1,\ E_{CO2}INC >=0 \text{ ------------}\mathbf{(8.2)} \end{aligned}$$

From (3.1) –

If k=0, then $P_{consmp}ITC = 0$, So, $P_{ECO2}=0$

If k=1, then $P_{consmp}ITC = 0$, So, $P_{Pconsmp}= Task_{Pconsmp}$

If k=2, then $P_{Pconsmp}= 2Task_{Pconsmp} + P_{consmp}ITC$

From (3.2) –

If k=0, then $E_{CO2}ITC = 0$, So, $P_{ECO2}=0$

If k=1, then $E_{CO2}ITC = 0$, So, $P_{ECO2} = Task_{Eco2}$

If k=2, then $P_{ECO2} = 2Task_{Eco2} + E_{CO2}ITC$

From (5.1)

If p = 0, then $P_{consmp}IPC$, So, $PSU_{Pconsmp} = 0$

If p = 1, then $P_{consmp}IPC = 0$, So, $PSU_{Pconsmp} = P_{Pconsmp}$

If p = 2, then $PSU_{Pconsmp} = 2P_{Pconsmp} + P_{consmp}IPC$

From (5.2)

If p = 0, then $E_{CO2}IPC = 0$, So, $PSU_{ECO2} = 0$

If p = 1, then $E_{CO2}IPC = 0$, So, $PSU_{ECO2} = P_{ECO2}$

If p = 2 then $PSU_{ECO2} = 2P_{ECO2} + E_{CO2}IPC$

From (8.1)

If n = 0, then $P_{consmp}INC = 0$, So, $PSU_{MN\_Pconsmp} = 0$

If n = 1, then $P_{consmp}INC = 0$, So, $PSU_{MN\_Pconsmp} = PSU_{Pconsmp}$

If n = 2, then $PSU_{MN\_Pconsmp} = 2PSU_{Pconsmp} + P_{consmp}INC$

From (8.2)

If n = 0, then $E_{CO2}INC = 0$, So, $PSU_{MN\_ECO2} = 0$

If n = 1, then $E_{CO2}INC = 0$, So, $PSU_{MN\_ECO2} = PSU_{ECO2}$

If n = 2, then $PSU_{MN\_ECO2} = 2PSU_{ECO2} + E_{CO2}INC$

Hence, we conclude that for a single task or a single process execution no inter task or inter process communication is required to minimize the power consumption and Carbon emission. Also we have shown that the power consumption and carbon emission due to execution of all processes submitted by a particular user on a single node is less than the multi node processing because of its inter node communication.

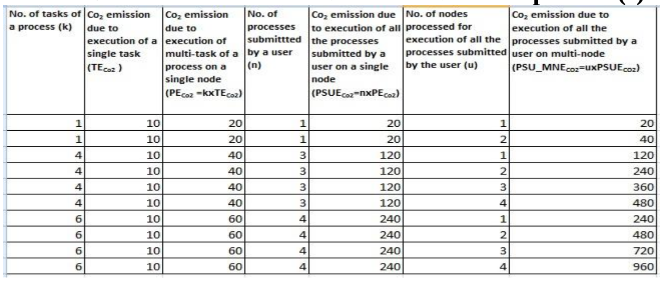
So, for small number of processes (similar sizes), if we avoid the inter process communication as well as the inter node communication to minimize the power consumption and carbon emission, i.e., $\mathbf{PSU_{MN\_Pconsmp}} = (n\ x\ PSU_{Pconsmp}) = (n\ x\ p\ x\ P_{Pconsmp})$ and $\mathbf{PSU_{MN\_ECO2}} = (n\ x\ PSU_{ECO2}) = (n\ x\ p\ x\ P_{ECO2})$

Once the EVALUATE module calculate the Power consumption & Carbon dioxide emission of a process, it can easily calculate the total Power consumption and CO2 emission, because at the time of process submission by the user EVALUATE module keeps track of the number of processes submitted by the user.

B. Experimental Setup

Since it is difficult to access actual data centers or cloud infrastructures, we used simulation-based tests that can be easily replicated to equate the performance of the proposed algorithm with the existing works that most cloud service providers are currently using.

Table I: Carbon Emission due to execution of process(s)



No. of tasks of a process (k)	Co_2 emission due to execution of a single task (TE_{Co2})	Co_2 emission due to execution of multi-task of a process on a single node ($PE_{Co2} = kxTE_{Co2}$)	No. of processes submitttted by a user (n)	Co_2 emission due to execution of all the processes submitted by a user on a single node ($PSUE_{Co2}=nxPE_{Co2}$)	No. of nodes processed for execution of all the processes submitted by the user (u)	Co_2 emission due to execution of all the processes submitted by a user on multi-node ($PSU\_MNE_{Co2}=uxPSUE_{Co2}$)
1	10	20	1	20	1	20
1	10	20	1	20	2	40
4	10	40	3	120	1	120
4	10	40	3	120	2	240
4	10	40	3	120	3	360
4	10	40	3	120	4	480
6	10	60	4	240	1	240
6	10	60	4	240	2	480
6	10	60	4	240	3	720
6	10	60	4	240	4	960

C. Analysis

The basic operational carbon emission (unit per second) of a task (based on the task size) is specified by a Serviced Provider or once calculate the carbon emission of a task based on the current work load consider the carbon emission of the task is fixed for that moment.

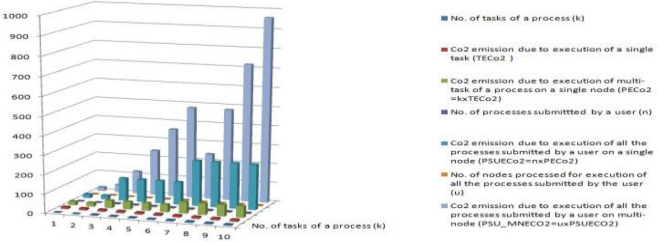


Fig. 4.Carbon emission vs. node execution

Our EVALUATE module of VTAMMS model keeps track of the number of submitted tasks of a process by a cloud user, so overall carbon emission can easily be calculated and it founds that total carbon emission varies with the number of nodes execution. From the above figure (Fig. 4) it is found that total carbon emission of all processes (submitted by the user) on a single node execution is much lower than on multimode execution.

VII. CONCLUSION

We have offered our novel approach to EVaaS, which considers VM user constraints along with physical host load factor to address the issue of mapping user's tasks into physical nodes in such a way that the number of nodes used is minimized, over-use and under-use of PM resources can be identified and resolved at the same time without violating any SLA agreements. Since we consider this as a environmentally viable Service that not only minimizes power consumption and carbon emission but serves as a mediator between authorized users and cloud service providers, i.e. without the permission of the EVSP, no one can communicate with each other. It may avoid the inequity between the actual consumption of electricity by the users and the billing records provided by the providers, so avoid any false accusations that may be claimed against each other in order to obtain illegal compensation. Based on the analysis, we have shown that our proposed algorithm uses a minimum number of physical machines to host a set of VMs, which also reduces the power consumption and carbon emission of the data center at a higher resource utilization rate.

VIII. FUTURE SCOPE

Our goal is to enrich the QoS, end-user utility, satisfactory user level as well as service providers to the extent that low investment costs are possible, and in the future we will try to implement how to clean the environment to avoid typical carbon emission impacts, which will make cloud computing more accessible.

ACKNOWLEDGEMENT

I would like to express my wholehearted gratitude and heartfelt sentiment to Dr. Debabrata Sarddar, my esteemed research advisor, for sparing his precious time and assimilating new, flawless concepts at every point of our work. I would not have been able to accomplish this research work effectively without his kind cooperation, encouragement and careful guidance.

REFERENCES

1. Himadri Biswas, Debabrata Sarddar, "Verification as a Service (VaaS): A Trusted Multipurpose Service for Accounting, Billing Verification and Approval of Consumed Resources at Low Computational Overhead to Enhance the Cloud Usability", International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 19 (2018) pp. 14402-14410 © Research India Publications. http://www.ripublication.com
2. Rajesh Bose, Himadri Biswas, Debabrata Sarddar , Manas Kumar Sanyal, "Cloud Billing & Verification Of Consumed Resources and Storage Spaces by a Cloud User" in International Journal of Applied Engineering Research ISSN 0973-4562 Volume 11, Number 9 (2016) pp 6568-6576.
3. P.Mell, T.Grance, "The NIST Definition of Cloud Computing,"September 2011. [Online]. Available: csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf. [Accessed10 January 2017].
4. F.lue, J.Tong, J.Mao, R.Bohen, J.Messina, L.Badger, D.Leaf, "NISTCloud Computing Reference Architecture," [Online]. Available: ws680.nist.gov/publication/get_pdf.cfm?pub_id=909505. [Accessed 10January 2017].
5. A.Khosravi, S.Kumar Garg, R.Buyya, "Energy and Carbon-EfficientPlacement of Virtual Machines in Distributed Cloud Data Centers," in
6. Euro-Par 2013 Parallel Processing, Springer Berlin Heidelberg, 2013,pp. 317-328.
7. M.Uddin, A.Abdul Rahman, "Energy efficiency and low carbon enablergreen IT framework for data centers considering green metrics," Renewable and Sustainable Energy Reviews, vol. 16, no. 6, p. 4078–4094, 2012.
8. S.Kumar Garg, C.Shin Yeo, R.Buyya, "Green Cloud Framework for Improving Carbon Efficiency of Clouds," in Euro-Par 2011 Parallel Processing, France, 2011, pp. 491-502.
9. F.Farrahi Moghaddam, R.Farrahi Moghaddam, M.Cheriet, "CarbonMetering and Effective Tax Cost Modeling for Virtual Machines," in
10. 2012 IEEE 5th International Conference on Cloud Computing(CLOUD), 2012.
11. A.Kansal, F.Zhao, J.Liu, N.Kothari, A.Bhattacharya, "Virtual machinepower metering and provisioning," in Proceedings of the 1st ACMsymposium on Cloud computing, Indianapolis, 2010.
12. F.Farrahi Moghaddam, M.Cheriet, K.Nguyen, "Low Carbon VirtualPrivate Clouds," in 2011 IEEE International Conference on CloudComputing (CLOUD), 2011.
13. J.Doyle, D.O'Mahony, R.Shorten, "Server selection for carbon emissioncontrol," in GreenNets '11 Proceedings of the 2nd ACM SIGCOMMworkshop on Green networking, 2011.
14. S.Mustafa, K.Bilal, S.A.Madani, N.Tziritas, S.U. Khan, L.T. Yang,"Performance Evaluation of Energy-aware Best Fit Decreasing
15. Algorithms for Cloud Environments," in 2015 IEEE InternationalConference on Data Science and Data Intensive Systems (DSDIS), 2015.
16. N.Quang-Hung, P.Dac Nien, N.Hoai Nam, N.Huynh Tuong, N.Thoai,"A Genetic Algorithm for Power-Aware Virtual Machine Allocation in Private Cloud," in Information and Communication Technology, Springer Berlin Heidelberg, 2013, pp. 183-191.
17. P.Xiang Gao, A.R.Curtis, B.Wong, S.Keshav, "It's not easy being green,"

in Proceedings of the ACM SIGCOMM 2012 conference on Applications, technologies, architectures, and protocols for computer communication, Helsinki, 2012.

18. F.Kong, X.Liu, "A Survey on Green-Energy-Aware Power Management for Datacenters," ACM Computing Surveys (CSUR), vol. 47, no. 2, 2015.
19. . R.Jansen, P.R.Brenner, "Energy efficient virtual machine allocation in the cloud," in Green Computing Conference and Workshops (IGCC),IEEE, 2011.
20. Y.Wu, M.Tang, W.Fraser, "A simulated annealing algorithm for energy efficient virtual machine placement," in 2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Seoul, 2012.
21. M.K.Patterson, S.W.Poole, C.Hsu, D.Maxwell, W.Tschudi, H.Coles, D.J.Martinez, N.Bates, "TUE, a New Energy-Efficiency Metric Applied at ORNL's Jaguar," in Supercomputing, Springer Berlin Heidelberg, 2013, pp. 372-382.
22. T.Wilde, A.Auweter, M.K.Patterson, H.Shoukourian, H.Huber, A.Bode, D.Labrenz, C.Cavazzoni, "DWPE, a new data center energy-efficiency metric bridging the gap between infrastructure and workload," in 2014 International Conference on High Performance Computing & Simulation (HPCS), 2014.
23. T.Samizadeh Nikoui, S.Jabbehdari, A.Bagheri, "Providing a Cloud Broker based Approach to Improve the Energy Consumption and Achieve a Green Cloud Computing," International Journal of ComputerApplications, vol. 138, pp. 42-49, 2016.
24. . H.F.Hamann, V.López, "Data Center Metrology and Measurement Based," in Energy Efficient Thermal Management of Data Centers, Springer US, Springer US, pp. 273-334.
25. M.K.Patterson, "Energy Efficiency Metrics," in Energy Efficient Thermal Management of Data Centers, Springer US, 2012, pp. 237-271.
26. D.Cavdar, F.Alagoz, "A Survey of Research on Greening Data Centers," in 2012 IEEE Global Communications Conference (GLOBECOM), 2012.
27. A.Capozzoli, M.Perino, G.Serale, "Review on Performance Metrics for Energy Efficiency in Data Center: The Role of Thermal Management," in Energy Efficient Data Centers, Springer International Publishing, 2015, pp. 135-151.
28. J.Yuventi, R.Mehdizadeh, "A critical analysis of Power Usage Effectiveness and its use in communicating data center energy consumption," Energy and Buildings, vol. 64, pp. 90-94, 2013.
29. Li.Wang, S.U.Khan, "Review of performance metrics for green data centers: a taxonomy study," The Journal of Supercomputing, vol. 63, no. 3, p. 639–656, 2013.
30. C.Dupont, "Building Application Profiles to Allow Usage of the Renewable Energies in Data Centres," in Energy Efficient Data Centers , Springer International Publishing, 2015, p. 120–131.
31. B.Kaur, A.Kaur, "An Efficient Approach for Green Cloud Computing," in 2015 1st International Conference on Next Generation Computing Technologies (NGCT), 2015.
32. . A.C.Adamuthe; R.M.Pandharpatte; G.T.Thampi, "Multiobjective Virtual Machine Placement in Cloud," in 2013 International Conference on Cloud & Ubiquitous Computing & Emerging Technologies, 2013.
33. J.Yuventi, R.Mehdizadeh, "A critical analysis of Power Usage Effectiveness and its use incommunicating data center energy consumption," Energy and Buildings, vol. 64, p. 90–94, 2013.
34. B.Wadhwa, A.Verma, "Carbon efficient VM placement and migration technique for green federated cloud datacenters," in 2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI, 2014.
35. S.Khalil Abd, S.A.R Al-Haddad, F.Hashim, A.B.H.J.Abdullah , S.Yussof,, "An effective approach for managing power consumption in cloud computing infrastructure," Journal of Computational Science, 2016.
36. I.Kar, R.N.Ramakant Parida, H.Das, "Energy aware scheduling using genetic algorithm in cloud data centers," in 2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT), 2016.
37. E.Feller; L.Rilling; Ch.Morin, "Energy-Aware Ant Colony Based Workload Placement in Clouds," in 2011 IEEE/ACM 12th International Conference on Grid Computing, 2011.
38. E.Feller, Ch.Morin, A.Esnault, "A case for fully decentralized dynamic VM consolidation in clouds," in 2012 IEEE 4th International Conference on Cloud Computing Technology and Science (CloudCom), 2012.
39. .Himadri Biswas, Debabrata Sarddar, "Power Management as a Service (Pmaas) – An Energy Saving Service by ―Auto-Fit VM Placement‖ Algorithm in Cloud Computing for Maximum Resource Utilization at Minimum Energy Consumption Without Live-Migrating the Virtual Machines by using the Concept of Virtual-Servers" in International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-9 Issue-2, December 2019.
37. Debabrata Sarddar, Himadri Biswas and Priyajit Sen, "Safety as a Service (SFaaS) Model - The New Invention in Cloud computing to establish a Secure Logical Communication Channel between Data Owner and the Cloud Service Provider before Storing, Retrieving or Accessing any Data in the Cloud" International Journal of Grid and Distributed Computing, ISSN 2005-4262 Vol. 10, No. 10 (2017), pp.1-20

AUTHORS PROFILE

Himadri Biswas, formar Assistant Professor at the Department of Computer Science & Engineering, Modern Institute of Engineering & Technology, Bandel, Hooghly, West Bengal, INDIA. He also has 16 years+ of teaching experience and has served in several institutions. He is currently pursuing a Ph.D. degree from the University of Kalyani, Nadia, West Bengal, India. He's done his M. Tech in Computer Science & Engineering from WBUT in 2010 and his MCA from St. Xavier's College in Kolkata under IGNOU in 2004. His research interests include cloud computing, mobile computing.

Dr. Debabrata Sarddar, Assistant Professor at the Department of Computer Science and Engineering, University of Kalyani, Kalyani, Nadia, West Bengal, INDIA. He holds a Ph.D. degree at the University of Jadavpur. He completed his M.Tech in Computer Science & Engineering from DAVV, Indore in 2006 and his B.E. in Computer Science & Engineering from NIT, Durgapur in 2001. He published about 200 research papers in various journals, attended 50 conferences, wrote 10 book chapters and 3 books. His research interests include wireless and mobile systems, cloud computing and the wireless sensor network.

Uddyalok Chakraborty, Assistant Teacher at the Department of Computer Science, Sharda Mandir H.S. School under Banasthali Vidyapith (Deemed University), Banasthali, Tonk, Rajasthan, India. He has 2 years of teaching experience and also 1 year of Industrial experience. He has completed his B.Tech in Computer Science & Engineering from MAKAUT in 2018. His research interests include cloud computing, mobile computing.