Analysis of Cloud Network Management Using Resource Allocation and Task Scheduling Services

[1]K.C. Okafor
[1]Dept. of Electrical and Electronic Engineering, Federal University of Technology, Owerri, Nigeria

[2]F.N.Ugwoke
[2]Dept. of Computer Science, Michael Okpara University of Agriculture, Umudike Umuahia, Nigeria

[3]Obayi, Adaora Angela
[3]Dept. of Computer Science, University of Nigeria, Nsukka Nigeria

[4]V.C Chijindu, [5]O.U Oparaku
[4&5]Dept. of Electronic Engineering, University of Nigeria, Nsukka, Nigeria

Abstract—Network failure in cloud datacenter could result from inefficient resource allocation; scheduling and logical segmentation of physical machines (network constraints). This is highly undesirable in Distributed Cloud Computing Networks (DCCNs) running mission critical services. Such failure has been identified in the University of Nigeria datacenter network situated in the south eastern part of Nigeria. In this paper, the architectural decomposition of a proposed DCCN was carried out while exploring its functionalities for grid performance. Virtualization services such as resource allocation and task scheduling were employed in heterogeneous server clusters. The validation of the DCCN performance was carried out using trace files from Riverbed Modeller 17.5 in order to ascertain the influence of virtualization on server resource pool. The QoS metrics considered in the analysis are: the service delay time, resource availability, throughput and utilization. From the validation analysis of the DCCN, the following results were obtained: average throughput (bytes/Sec) for DCCN = 40.00%, DCell = 33.33% and BCube = 26.67%. Average resource availability response for DCCN = 38.46%, DCell = 33.33%, and BCube = 28.21%. DCCN density on resource utilization = 40% (when logically isolated) and 60% (when not logically isolated). From the results, it was concluded that using virtualization in a cloud DataCenter servers will result in enhanced server performance offering lower average wait time even with a higher request rate and longer duration of resource use (service availability). By evaluating these recursive architectural designs for network operations, enterprises ready for Spine and leaf model could further develop their network resource management schemes for optimal performance.

Keywords—Resource Provisioning; Virtualization; Cloud Computing; Service Availability; Smart Green Energy; QoS

I. INTRODUCTION

A. Background Study

Middleware solutions for heterogeneous distributed cloud datacenters aim to respond to high requirements of large scale distributed applications relating to performance, flexibility, portability, availability, reliability, trust and scalability in the context of high number of users. These are usually considered in large geographic distribution of heterogeneous hardware and software resources. The concepts used in the design, implementation, and deployment of systems with such capabilities could be based on demand side management and monitoring, optimization via scheduling, sharing, load balancing, consolidation and other high performance grid based techniques. In most cases, new services and functionalities could be added to the middleware to enhance data-intensive and highly demanding applications with low cost and high performance. New cloud computing architectures must be designed to incorporate solutions for the management of data, resources, tasks, and applications. It must ensure fault tolerance, accounting, service on demand, and other functions required by user communities to operate effectively in a shared services environment.

These observations formed the philosophy of an on-going research known as Smart Green Energy Management System (SGEMS). The system is a renewable energy system based on Solar PV microgrid, cloud energy meter and Distributed Cloud Computing Network (DCCN). In this system, the cloud datacenter server acts as the supporting platform for Enterprise Energy Tracking Analytic Cloud Portal (EETACP) deployment. But as energy users send their job tasks, fairness must be maintained optimally. Fairness in context refers to the method of having each job receive equal (or weighted) share of computing resources at any given moment. The DCCN must satisfy the fairness criteria for EETACP workload in the SGEMS research.

Studies have shown that datacenters are now the enterprise foundations that support many Internet applications, enterprise operations, and novel scientific computations like cloud computing services for distributed energy management platforms. In fact, they are large-scale data-intensive computing infrastructure. The major challenge facing smart green IT researchers is how to build a scalable cloud based DCN platform that delivers significant aggregate bandwidth and excellent Quality of Service (QoS) for smart grid web platforms. On this issue, research efforts such as Fat-tree [1], [2], VL2 [3], Monsoon [4], DCell [5], MDCube[6], BCube [7], FiConn [8], DPillar[9], DRweb[10], SVLAN [11], and Scafida [12], etc, have been proposed in recent years based on their switch and server-centric network architectures, with no attention on excellent resource management schemes. However, these works have made significant contributions on server interconnectivity primarily. For fairness in this type of

network, resource allocation and scheduling remains indispensable.

According to [13], by running large-scale computation and data-intensive services on inexpensive server clusters and other large-scale data parallel systems, cloud provisioning, (i.e., allocating resources for cluster requests) remains key to consolidating such clusters. Basically, resource management problems in multi-cluster environments are broadly classified into three large categories viz:

1) Cloud providers provisioning/delivering raw clusters based on resource requirements of their customers.

2) Customers running cluster operating systems to manage critical server resources and schedule jobs from multiple frameworks.

3) Tasks Scheduling frameworks with or without assistance from the cluster operating system to get the job done.

The fundamental goal of a well-developed resource management scheme on a server cluster is to create a cost effective model taking cognizance of the aforementioned categories while formulating a validation mechanism that will justify performance of the proposed system. The SGEMS EETACP platform that runs on the DCCN places a computation demand on resource allocation in cloud based environment. Since stability criterion must be satisfied, the use of virtual machine as the minimum resource allocation unit in the DCCN could suffice. When a user starts an application, a virtual machine that satisfies the minimum resource requirement for the application is allocated via scheduling map. When workload of the application increases as a result of user traffic, a new virtual machine is allocated for this application. This must allocate more physical resources (CPU, Memory, etc.) to the existing virtual machine, without shutting down the existing virtual machine for resource reallocation. This is most ideal for EETACP as a mission critical application since downtime is not an option or desired.

To provide guarantees of server operations, the datacenter with a cluster of servers must provision sufficient resources to meet application needs. Such provisioning can be based either on a dedicated or a shared model. In the dedicated model, a number of cluster nodes are dedicated to each application and the provisioning technique must determine how many nodes to allocate to the application [14]. In the shared model, running cloud virtualization for resource scheduling can allow running applications to share server resources with other applications. Once the cloud driver allocates a set of resource units such as virtual machines, the MapReduce system uses the resources that are heterogeneous shared among multiple jobs in EETAACP context.

Therefore, this paper focuses on studying resource allocation and scheduling at both the application level and the backend core to see how to map the physical resources to Virtual machines for better resource utilization in DCCN environment.

B. Contributions

In this work, an experimental investigation on metrics associated with job scheduling and resource allocation on a shared heterogeneous server cluster was carried out using DCCN, DCell and BCube as a case studies. Virtual Machine (*Vm*) algorithms were developed to provide good performance while guaranteeing fairness in an operational setup. Consequently, this will represent the ideal mode for EETACP service provisioning. The perspective offered in context is that an efficient task resource scheduling algorithm based on virtualization should be implemented at the broker domain for the DCCN. This was carried out to facilitate the deployment of a previous work on EETACP service proposed for SGEMS in an earlier work. The aim is to dynamically allocate the virtual resources in the EETACP as well as other services based on their workload intensities. This is to improve resource utilization, throughput, and availability and reduce the usage cost.

The rest of this paper is organized as follows. Section II presented the literature review as well as foundational concepts. Section III discussed the methodology and relevant system Models. Section IV presents the system Validation from the simulation environment. Section V concludes the paper.

II. LITERATURE REVIEW

In this section, two interrelated concepts will be clarified so as to provide a working foundation for EETACP deployment in DCCN at large. These are virtualization and resource allocation. Secondly, this work will then present the related works.

A. Cloud Virtualization and Resource Allocation

In [15], Virtualization is defined as the mirror imaging of one or more workstations/servers, etc within a single physical computer utilizing the same system resources. Virtualization makes cloud computing possible, since scalability is the major consideration in cloud computing. Cloud computing servers use the same operating systems, enterprise and web applications as localized virtual machines and physical servers [16]. Other views on the concept are detailed in [17] and [18]. Other the other hand, resource allocation is the process of assigning available resources to the needed cloud applications over the internet via the cloud DataCenter. A dynamic resource allocation framework (resource controller) in cloud environment helps to monitor traffic load changes, analyse workload and facilitate the implementation of an automated elastic resource controller that ensures high availability.

Resource controller in context, controls all the components on the cloud side and it has access to the load balancer, monitoring data, and front end (open nebula) for requesting additional resources on demand [19]. Some vendor services on a distributed cloud may include: computational resource configuration of the Virtual Machines (VMs), the programmer's degree of control, network service configuration, the nature of hardware/software security services, portability guarantees, storage scalability, etc, as such there is a need for a comprehensive resource allocation and scheduling system for cloud datacenter networks (CDCN) [19].

It is worthy to note that the allocation of resources to dedicated servers without virtualization schemes could be

problematic, while over-provisioning resources based on worst case workload estimates can result in potential network crash/failure and then violate of guarantees of QoS. An alternate approach is to allocate resources to servers dynamically based on the variations in user workloads profiles. In this approach, each server is given a certain minimum share based on coarse-grain estimates of its user resource needs.

Such dynamic resource sharing can yield potential multiplexing gains, while allowing the system to react to unanticipated increases in application load thereby meeting the QoS guarantees if both are cloud based.

In addition, for an excellent resource provisioning technique, there is need to determine the influence of service availability and processing delays on the server backend. By sharing server resources, this can provide guarantees to applications in the cloud DataCenter model. However, such guarantees are provided by reserving a certain fraction of node resources (CPU, network, and disk) for each application. In this regard, the size of the resources allocated to each cloud server will depend on the expected workload and the QoS requirements of the application.

For these workloads, there is a need to actually ascertain the influence of resource allocation using virtualization in enterprise computing cloud applications. Knowing the kind of servers that will scale in the event of high traffic density is very vital.

Consequently, this paper will use the concept of virtualization to explain the resource allocation and scheduling features in a cloud based management platform. The emphasis is on user jobs (workload) on server pools for a proposed DCCN. In developing this paper, strong emphasis is placed on dynamic resource allocation and scheduling technique via virtualization so as to handle changing application workloads in shared distributed Cloud Computing backend environment.

B. Related Research Efforts

Several works have been carried out on resource allocation and scheduling focusing on cloud infrastructure enhancement. This work will discuss these efforts below leveraging the Systematic Literature Review (SLR) approach.

The authors in [20] classified resource allocation models into three categories viz: processing resources, network resources and energy efficient resources. The work opined that network performance and resource availability could pose the tightest bottleneck for any cloud platform. Traditional Resource Management Systems (RMSs) such as Condor [21], LoadLeveler [21], Load Sharing Facility (LSF) [22], and Portable Batch System (PBS) [168], all adopt system-centric resource allocation approaches that focus on optimizing overall cluster performance. However, these have not been exploring in Spine leaf DCCN. In [24], a SLA-oriented resource management system built using Aneka [170] for cloud computing was proposed. A representative sample of works on resource allocation in the cloud environment for job task processing, and other resource categories is detailed in [25], [26], [27], [28], [29] and [30]. These works failed to justify their relevance in Spine leaf DCCN. Also, a process

scheduling and its algorithm was presented in [31],[32]. The works considered a resource allocation scheme on multiple clouds in cases of under load and the over load conditions. The paper in [33] a proposed model for cloud computing scheduling based on multiple queuing models for improve the quality of service by minimize execution time per jobs, waiting time and the cost of resources to satisfy user's requirements. The experimental results indicate that the model increases utilization of global scheduler and reduce waiting time. But, achieving resource allocation in a DCCN is only feasible through scheduling as a DCCN service. Similarly, the authors in [33], established some scheduling schemes and their strategies which were explained in [34]. The opined that both cannot be used in cloud computing for Application Processing Requests (APR) as found in [35],[36] owing to some identified QoS limitations. In [37], the authors proposed a cost-optimal scheduling in hybrid IaaS due to divergent users' requirements and heterogeneous workload characteristics.

The authors observed that the problem of scheduling a user's workload in the cloud remains a complex thereby proposing an optimal cost scheduling scheme. In [38], a genetic algorithm scheduling approach was proposed for addressing the problems of scheduling with traditional algorithms resulting in load imbalance and high migration costs. Other efforts made in literature in these areas of resource scheduling include: Greedy Particle Swarm Optimization (GPSO) [39], Task Length and User Priority (ie. Credit Based Scheduling Algorithm) [40], Cost based scheduling [41], Energy efficient optimization methods [42], Activity based costing [43],[44], Reliability Factor Based [45], Context aware scheduling [46],Dynamic slot based scheduling [47],[48], Multi-Objective Tasks Scheduling Algorithm [49], Public Cloud Scheduling Algorithm with Load Balancing [50], Agent-based elastic Cloud bag-of-tasks concurrent scheduling [51], Analytic hierarchy process (task scheduling and resource allocation) [52], Swarm scheduling [53], Profit-driven scheduling [54], Dynamic trusted scheduling [55], Community-aware scheduling algorithm [56], Adaptive energy-efficient scheduling [57], grid, cloud and workflow scheduling [58]. In these algorithms, job/task length and priority are mostly the parameters analyzed. However, the SGEMS DCCN inherits from the characteristics of the above works, but focused on improving the network QoS with respect to state of art Spine Leaf network model. The research gaps below were used to conclude the findings from literature review above.

C. Research Gaps

The following were clearly identified from literature study.

- QoS Resource Management

From the literature reviews, it has been shown that resource allocation, scheduling and service provisioning are the critical concepts in DataCenter network operations which must be considered when designing an efficient cloud based network. But existing works have not resolved the issues of excellent quality of service in cloud servers via virtualization scheme particularly for DCCN running on a spine leaf operational mode.

Consequently, this work proposes a dynamic architecture that handles all the resources in the DCCN by managing client requests, directing resource allocation, eliminating performance constraints, minimizing cost while ensuring the overall QoS. In this paper, resource management for client requests is carried out in the server cluster pool

- Validation Comparison with DCell and BCube

Based on the heuristic branch-and-bound concept with Riverbed modeller, a scenario based study with similar network architectures (DCell and BCube) will be carried out considering throughput, resource availability and network density as metrics. To the best of our knowledge, this work is the only work that have carried out a scenario based comparison with scalable DCell and BCube on the basis of heuristic tasks and priority scheduling in spine leaf DCCN.

III. RESEARCH METHODOLOGY

The method used in this work is referred to procedural benchmarking with Riverbed Modeller 17.5. In this case, a step by step approach was employed in studying BCube and DCell legacy x86 server consolidations as typified in University of Nigeria Nsukka (UNN) DCN as a case study. With the identified QoS issues in the network, this work leveraged server virtualization based on VMware vSphere which is a more mature and trusted technology in the enterprise spine leaf DCCN. In this regard, this work considered three key conditions when applying parallel processing in executing tasks in DCCN server viz: 1) how to allocate resources to tasks; 2) in what order is the task executed in the cloud; and 3) how to schedule overheads when VMs prepare, terminate or switch tasks. Task scheduling and resource allocation basically address these issues. The procedural benchmarking approach used in this work took care of the initial design specification and composite process model of the DCCN. This model architecture is presented next

A. DCCN Model Archicture/Specifications

Considering the DCCN architecture shown in Fig 1. The design comprises of two layers functional areas, viz: remote user access and the hybrid speed redundancy layer. The gateway load balancer (GLB)/speed redundancy layer was used interchangeably with the Integrated Service OpenFlow load balancer (ISOLB) in this work. The ISOLB connects the cloud layer to the broker which coordinates the *Vms.*

The cloud computing architecture in Fig.1 uses the cloud broker to mediate negotiations between EETACP Software as a Service (ESaaS) and cloud provider. This negotiation is driven by QoS requirements. The broker acts on behalf of ESaaS for allocation of resources that can meet application's QoS requirements. In the DCCN, the ISOLB is the major component in the hybrid speed redundancy layer. However, the hybrid speed redundancy layer comprises the Virtual machines interconnected server subnet clusters and the ISOLB.

The architectural decomposition of DCCN will be discussed next while exploring its functionalities for grid performance using virtualization metrics. A generalized specification of the proposed DCCN datacenter is presented below.

- Let $DCCN_{lb}$ be an acronym chosen for the DCCN server cluster managed by the ISOLB controller. $DCCN_{lb}$ is designed to have various subnets for its clusters (eg. subnet *1* to *n*) referred to as $DCCN_{sa}$, $DCCN_{sb}$, $DCCN_{sc}$, $DCCN_{sd}$ interconnected together. $DCCN_{sa}$ represents a subnet as shown in Fig1. where *n* is a subnet factor such that *n> 0*. Each cluster ($DCCN_s$) uses High Performance Computing (HPC) servers running *Vm* with the ISOLB controller layered in linearly defined architecture. Since the design of datacenter network is for efficient server load balancing and EETACP application integration, the requirement of 4-ports from ISOLB controller and few servers necessitated the choice of four subnets. Virtual server instances running on the HPC servers expanded the server cluster capabilities.

- Servers in DCCN cluster are connected to ISOLB port of the load balancer corresponding to it, and owing to the running virtual instances Vi, a commodity 4-port switching/routing device with 40GB/s per port serve the design purpose. Also, each of the $DCCN_s$ is interconnected to each other through the ISOLB switch ports.

- The virtualized servers used in this work have two ports for redundancy (in Gigabytes). Each server is assigned a 2-tuple $[a_1, a_0]$ in consonance with its ports (a_1, a_0 are the redundant factors) together with an OpenFlow VLAN id.

- Emulated NEC IP8800 OpenFlow controller was the ISOLB used in this work; hence, the number K is the maximum number of OpenFlow-VLAN that can be created in it. The load balancer switch is a multilayer commodity switch that has a load balancing capability. This capability together with its OpenFlow VLAN capability was leveraged to improve the overall DCCN performance.

- Each server has its interface links in $DCCN_s$. One connects to an ISOLB, and other servers connect as well but all segmented within their subnets via OpenFlow-VLAN segmentation, as shown in Fig 2. OpenFlow DCN_s servers have virtual instances running on it and are fully connected with every other virtual node in the architecture.

Virtualization facilitates efficient use of hardware and software resources in DCCN. Hence, Virtual Machines *Vms,* are allocated to the user based on their job in order to reduce the number of physical servers in the cloud environment particularly in a high gird environment. But most VM resources are not efficiently allocated based on the characteristics of the job to meet out Service Level Agreements (SLA). Hence, this work introduced smart *Vm* allocation algorithm based on the characteristics of the EETACP job which can smartly reconfigure virtual resources thereby improving resource utilization in the server clusters.

Again, the DCCN port interface model for the ISOLB is shown in Fig.2. This creates advanced redundancy and multiplexing of job requests to the server clusters in the DCCN. For the DCell and BCube, these are not visible in their architecture. The Algorithm for *Vm* allocation and scheduling in a multiplexed server setup is shown in Algorithm I.
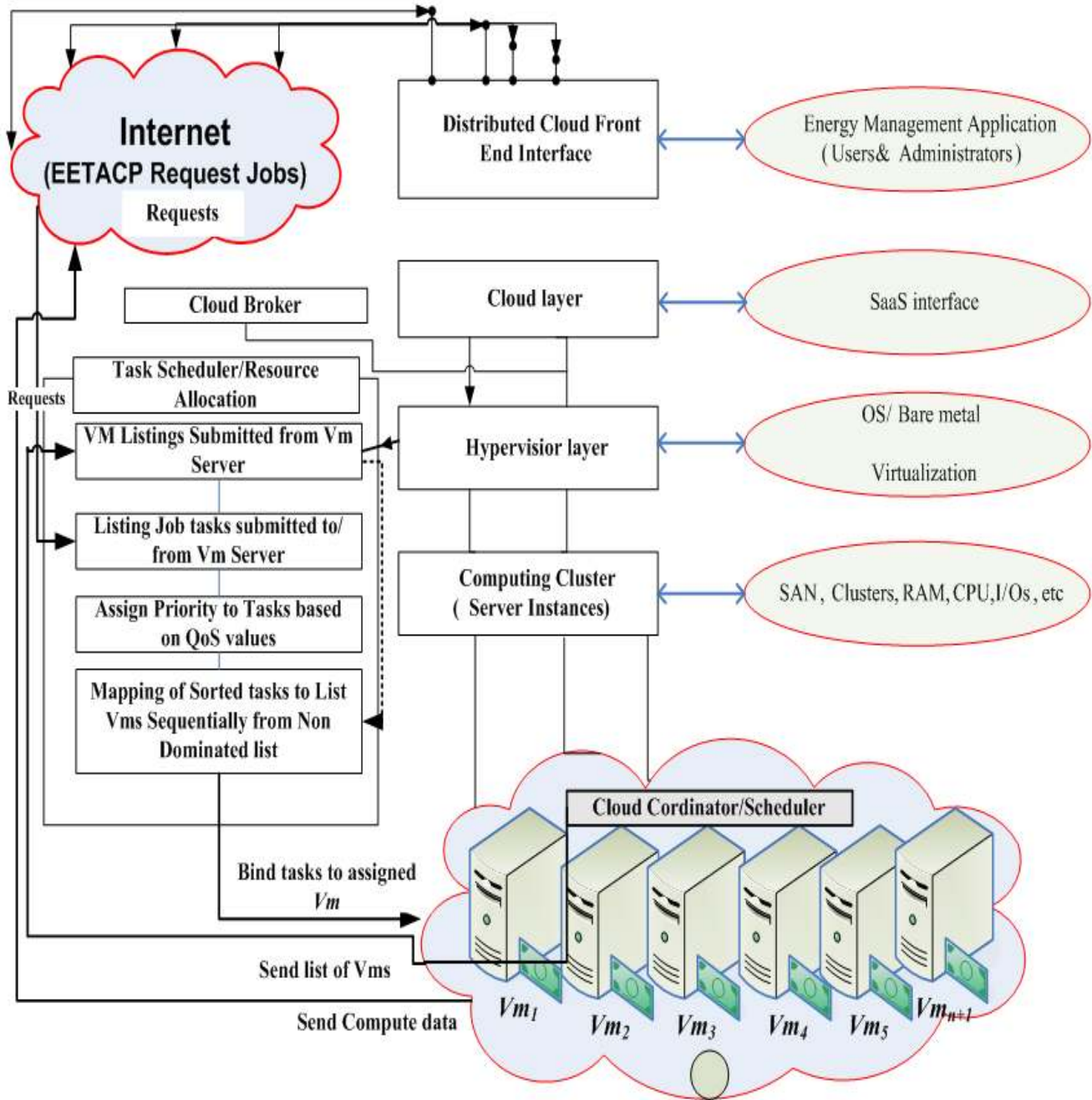

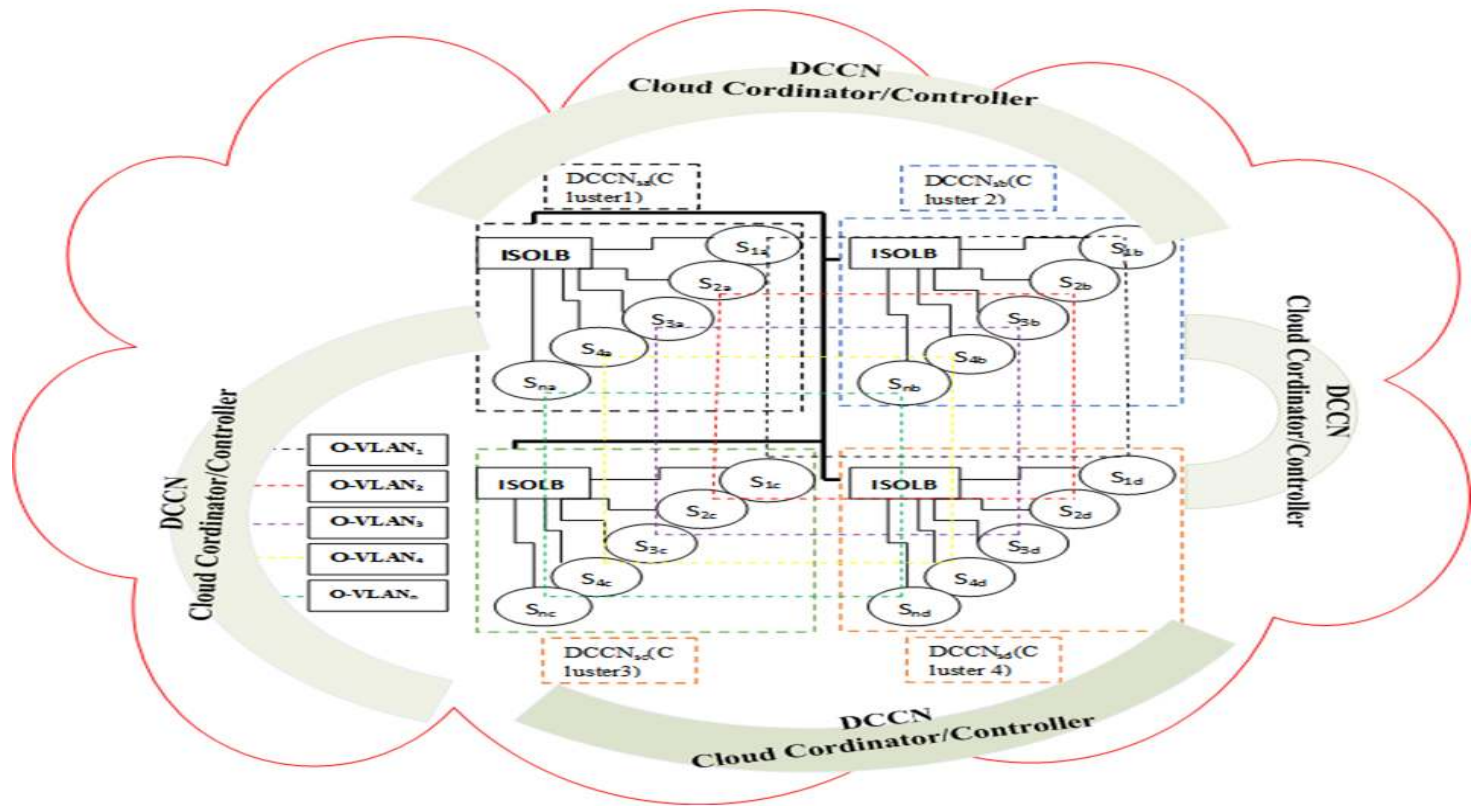Fig. 1. DCCN Resource Allocation and Task Scheduling with Virtual Machines

Fig. 2.   DCCN ISOLB logical Server Mapping with VLAN Service

Algorithm I shows this concept which was not found in DCell and BCube models.

Algorithm I: Execution of High priority job when all existing resources are allocated in DCCN Server Cluster

---

Input: New job, all jobs running in EETACP host
Output: Execution of all jobs submitted to the EETACP host
Process
*1 **Begin** ()*
2 Arrival of New job for user I into Fig 1
*3 **if** (New job.deadline< all jobs running in host)*
4 High priority job from user i→New job
*5 **if** (VM is available)*
6 allocate High priority job to that VM
*7 **else***
8 Suspend job ←Selection of Job for execution of high priority job();
9 Suspend (Suspend job)
10 allocate High priority job to VM from which a job was suspended
*11 **end if**;*
12 Execution of all jobs running in the VM Instance
*13 **if** (completion of a job which is running in VM)*
14 resume (Suspend job)
15 allocate the resumed job request to that VM Instance
*16 **End if***
17 Execution of resumed job in active state
*18   **End***

B. DCCN Service Cordinator

In DCCN, a virtualization algorithm known as DCCN logical construction algorithm was introduced to consolidate the efforts of the cloud coordinator controller in Fig.1 and Fig.2.

Recall that Fig 2 shows the design architecture for DCCN ISOLB logical server mapping with VLAN service. In this work, the ISOLB supports embedded VLAN service which offers excellent computing characteristics like flexibility, security, broadcast regulation, congestion control, etc. This is referred to as the Cloud Coordinator Service. It primarily isolates each server cluster from another, facilitates virtualization scheduling explained in Fig 1 and carries out resource allocation for EETACP workload in the cloud servers. The DCCN OpenFlow coordinator service algorithm is shown in Algorithm II.

The first section checks whether $DCCN_s$ server subnet cluster is constructed. If so, it connects all the server nodes n to a corresponding ISOLB port and ends the recursion. The second section interconnects the servers to the corresponding switch port and any number of $PS_K . Vm$ or P_i servers are connected with one link. Each server in the subnet cluster $DCCN_{lb}$ is connected with 40GB links for all OpenFlow VLAN*id*. The role of the service coordinator in the cloud is enormous. The DCCN logical architecture with the OpenFlow VLAN segmentation shown in Fig 2 uses the linear construction algorithm II for VLAN resource scalability.

Algorithm II: DCCN Service Coordinator OpenFlow VLAN Construction Algorithm.

/* l stands for the level of DCCN$_s$ subnet links, *n* is the number of nodes in a cluster
DCCN$_{lb}$,
pref is the network prefix of DCCN$_{lb}$s is the number of servers in a DCCN$_{lb}$cluster/
Build DCCN$_s$ (l, n, s)
Section I: / build DCCN$_s$ */*
    If *(l = = 0)* **do**
    For *(inti = 0; i< n; i++) /* where n is=4*/*
Connect node [pref, i] to its switch;
Return;
*Section II: /*build DCCN$_s$ servers*/*
For *(inti = 0; i< s; i++)*
Build DCCN$_s$ ([pref, i], s)
Connect DCCN$_s$ (s) to its switch;
 Return;
End

In the DCCN logical structure shown in Fig 2, the servers in one subnet are connected to one another through one of the ISOLB ports that is dedicated to that subnet. Each server in one subnet is also linked to another server of the same order in all another subnets. Incast collapse in cloud datacenter must be avoided at all cost. As such, each of the servers has two links, with one, it connects to other servers in the same subnet (intra server connection) and with the other it connects to the other servers of the same order in all other subnets (inter server connection). Apart from the communication that goes on simultaneously in the various subnets, the inter server connection is actually an OpenFlow VLAN connection. This OpenFlow VLAN segmentation of the servers logical isolates them for security and improved network performance. Together with other server virtualization schemes ultimately improves the performance in terms of throughput and other QoS metrics. The OpenFlow VLAN segmentation gave each DCCN$_s$ (subnet) the capacity to efficiently support enterprise web applications (EETACP, Web Portals, Cloud applications such as ESaaS) running on server virtualization in each port thereby lowering traffic density.

C. Coordinator Logical Isolation of Server Clusters

As shown in Fig.2, the application of OpenFlow VLAN in each subnet creates full logical isolation of the DCCN server cluster architecture of Fig 1. In order to achieve this, each server and nodes in DCCN$_s$ is assigned virtualization identity, $[V_{id} = av_1, av_2 \ldots\ldots av_{n-1}]$ and OpenFlow VLAN identity ($V_l$id) greater than 0, where $av_1, av_2 \ldots\ldots av_{n-1}$is the virtualization instances on DCCN$_s$ servers. As such each server can be equivalently identified by a unique $V_l$id in the range $V_l id \leq 1005$*.

Hence the total of $V_l$id for servers in the DCCN$_s$ is given by (1)

$$T (Vl_{id}) = \sum_{id=1}^{N} Vl_{id} * V_m(i, \phi) \qquad (1)$$

Where *N* is the maximum number of OpenFlow VLAN ids, and $V_m(i, \phi)$ is the virtual instances in the DCCN$_{lb}$ physical servers. The mapping between a unique $V_l$id and the DCCN$_{lb}$ physical servers considering that there are four subnet clusters DCCN$_{lb}$ in Fig 2 is given by (2)

$$DCCN_{lb\,Mapping} = 4 * Vl_{id} * V_m(i, \phi) \qquad (2)$$

Following the DCCN architecture in Fig.1, in order to minimize broadcast storms and reduce network traffic/demand density, an OpenFlow VLAN mapping scheme for the servers in the subnet clusters DCCN$_{lb}$was applied resulting to the system validation model discussed in section IV.

Now, consider DCCN$_{sa}$, DCCN$_{sb}$, DCCN$_{sc}$ and DCCN$_{sd}$ with servers $S_1$to S_n. The servers in each of the subnet cluster are mapped into different OpenFlow_VLAN services with their corresponding *ids* as follows:

$OpenFlow\_VLAN_1 \rightarrow S_{1a}, S_{1b}, S_{1c}, S_{1d}\ldots\ldots\ldots\ldots S_{1n}$
$OpenFlow\_VLAN_2 \rightarrow S_{2a}, S_{2b}, S_{2c}, S_{2d}\ldots\ldots\ldots\ldots S_{2n}$
$OpenFlow\_VLAN_3 \rightarrow S_{3a}, S_{3b}, S_{3c}, S_{3d}\ldots\ldots\ldots\ldots S_{3n}$
$OpenFlow\_VLAN_4 \rightarrow S_{4a}, S_{4b}, S_{4c}, S_{4d}\ldots\ldots\ldots\ldots S_{4n}$
$OpenFlow\_VLAN_n \rightarrow S_{na}, S_{nb}, S_{nc}, S_{nd}\ldots\ldots\ldots\ldots S_{nn}$
Where $S_{1a}, S_{2a}, S_{3a}, S_{4a}$ are the servers in DCCN$_{sa}$
$S_{1b}, S_{2b}, S_{3b}, S_{4b}$ are the servers in DCCN$_{sb}$
$S_{1c}, S_{2c}, S_{3c}, S_{4c}$ are the servers in DCCN$_{sc}$
$S_{1d}, S_{2d}, S_{3d}, S_{4d}$ are the servers in DCCN$_{sd}$.
With this OpenFlow VLAN mapping scheme, a logical isolation of the DCCN cluster subnets in the DCCN architecture was achieved. This make for smart flexibility, improved network security, agility and control of traffic flow in the DCCN core layer.

D. DCCN Resource Allocation

For user workloads in the virtualized server subnet cluster of Fig 2, by quantifying the workload requirements for the physical servers, the virtual machines can significantly benefit from the resource allocation strategies offered by the full virtualization scheme. It will be established that virtualization can offer excellent resource allocation of distributed server resources. In the DCCN, user tasks are assigned resources for effective performance in the server domain. The tasks represents user job request for server execution. I/Os, CPU, Memories are resources that are assigned in context. Resource allocation via virtualization makes for bandwidth availability, delay reduction and service availability in general. These are vital QoS metrics analyzed in this work.

From the framework of Fig.2,

Let n_n = User U_i task number or job numbers
   m_m= DCCN resource number,
   K_k= DCCN QoS parameter number
By introducing QoS parameters for each resource in a virtualized server, let's define a vector $V\beta_K^{res}$which gives the functional capabilities of a resource given in Equation (3).

$$\beta^{res} = \langle\beta_1^{res}, \beta_2^{res}, \beta_3^{res} \ldots\ldots\ldots\ldots\ldots \beta_K^{res}\rangle (3)$$

User U_i sends job request accompanied by a vector of QoS parameters, but the weights for the parameters are given in Equ 4 and 5 such that

$$\beta^{job} = \langle \beta_1^{job}, \beta_2^{job}, \beta_3^{job}, \dots\dots\dots\dots\dots, \beta_K^{job} \rangle \quad (4)$$

Vector weight $W_V = \langle W_1^{\grave{}}, W_2^{\grave{}}, W_3^{\grave{}}, \dots\dots\dots\dots\dots, W_k^{\grave{}} \rangle$

for $0 \le W_i^{\grave{}} \le 1$

Hence, $W_V = \sum_{i=i}^{k} W_i^{\grave{}} = 1 \quad (5)$

Here, each weight shows the importance of each parameter. Since CPU for instance is important for job execution, then it is activated or enabled by default. The same for other resources like I/Os, RAM, Storage disks, etc. If the resource provides solution for a requested service or task, the validity operator was introduced such that $R_{m_m} \to T_{n_n}$ in Equation (6)

$$\text{Hence, } R_{m_m} \to T_{n_n} = \left(\langle \sum_{i=0}^{k} \frac{\beta_1^{res_{m_m}}}{\beta_1^{job_{n_n}}} * W_{V_1} \rangle \right) \ge 1 \quad (6)$$

Where K = the number of the QoS Parameters.

From Equ 6, the resource specification for DCCN for Users U_i was optimized.

When the resource capability exceeds the task demand, then the virtualization function in (7) holds for the DCCN server.

$$F(R_{m_m} \to T_{n_n}) = \frac{\beta_1^{res_{m_m}}}{\beta_1^{job_{n_n}}} > 1 \quad (7)$$

To expand the optimization problem of assigning m_m resources to n_n tasks with K_k QoS parameters in the DCCN server cluster subnets, the compact matrixes was introduced in Equ 8, 9, and 10 for virtualization scheduling. These compact matrixes are described for resource allocation model in context. For users U_i, let task matrix requirements, be given by Equation (8).

$$T_{n_{n*K_K}} = \begin{pmatrix} \beta_1^{job_1} & \cdots & \beta_{k+1}^{job_1} \\ \vdots & \ddots & \vdots \\ \beta_1^{job_n} & \cdots & \beta_{k+1}^{job_n} \end{pmatrix} \quad (8)$$

For user U_i, let task weight matrix requirements be given by Equ 9

$$W_{n_{n*K_K}} = \begin{pmatrix} W_1^{job_1} & \cdots & W_{k+1}^{job_1} \\ \vdots & \ddots & \vdots \\ W_1^{job_n} & \cdots & W_{k+1}^{job_n} \end{pmatrix} \quad (9)$$

For users U_i, let server resource capability matrix requirements be given by Equ 10

$$R_{m_{m*K_K}} = \begin{pmatrix} \beta_1^{res_1} & \cdots & \beta_{k+1}^{res_{m_m}} \\ \vdots & \ddots & \vdots \\ \beta_k^{res_1} & \cdots & \beta_{k+1}^{res_{m_m}} \end{pmatrix} \quad (10)$$

By dividing (9) by (8) to obtain a Vm vector λ_{K_k}, to get Equation (11)

$$\lambda_{K_k} = \frac{W_{n_{n*K_K}}}{T_{n_{n*K_K}}} = [W_{n_{n*K_K}}][T_{n_{n*K_K}}]^{-1} \text{ such that}$$

$$\lambda_{K_k} = \begin{pmatrix} W_1^{job_1} / \beta_1^{job_1} & \cdots & W_{k+1}^{job_1} / \beta_{k+1}^{job_1} \\ \vdots & \ddots & \vdots \\ W_1^{job_n} / \beta_1^{job_n} & \cdots & W_{k+1}^{job_n} / \beta_{k+1}^{job_n} \end{pmatrix} \quad (11)$$

But resource allocation by V_m instance is given by (12),

$$V_{n_{n*M_m}} = \sum_{i=1}^{\phi} \left(\lambda_{K_k} * R_{m_{m*K_K}} \right) \quad (12)$$

Where ϕ = max virtual instance that can be accommodated by the physical server.

By substituting (10) and (11) into (12), a model for virtualization was developed by putting the task/job numbers (n_n), resource numbers (m_m) and QoS parameters (K_k) in (13)

$$V_{n_{n*M_m}} = \begin{pmatrix} \sum_{i=1}^{\phi} \left(\frac{W_1^{job_1}}{\beta_1^{job_1}} * \beta_1^{res_1} \right) & \cdots & \sum_{i=1}^{\phi} \left(\frac{W_1^{job_1}}{\beta_{K+1}^{job_1}} * \beta_1^{res_{m_m}} \right) \\ \vdots & \ddots & \vdots \\ \sum_{i=1}^{\phi} \left(\frac{W_1^{job_n}}{\beta_1^{job_n}} * \beta_1^{res_1} \right) & \cdots & \sum_{i=1}^{\phi} \left(\frac{W_1^{job_1}}{\beta_1^{job_n}} * \beta_1^{res_{m_m}} \right) \end{pmatrix} \quad (13)$$

Originally, it was established that the total server virtual instances in the DCCN server $PS_K.Vm$ is given by

$$PS_K.Vm = \sum_{i=1}^{m} V_m(i, \phi) = [V_{mi1} + V_{mi2} + \dots\dots\dots V_{mi_\phi})] \quad (14)$$

From the DCCN architecture shown in Fig 1, if the resource allocation model is given by Equ 13, then the consolidation model via virtualization for the physical servers can be obtained by substituting Equ 13 in Equ 14. This gives (15)

$$PS_K.Vm = \sum_{i=1}^{m} (1) * \begin{pmatrix} \sum_{i=1}^{\phi} \left(\frac{W_1^{job_1}}{\beta_1^{job_1}} * \beta_1^{res_1} \right) & \cdots & \sum_{i=1}^{\phi} \left(\frac{W_1^{job_1}}{\beta_{K+1}^{job_1}} * \beta_1^{res_{m_m}} \right) \\ \vdots & \ddots & \vdots \\ \sum_{i=1}^{\phi} \left(\frac{W_1^{job_n}}{\beta_1^{job_n}} * \beta_1^{res_1} \right) & \cdots & \sum_{i=1}^{\phi} \left(\frac{W_1^{job_1}}{\beta_1^{job_n}} * \beta_1^{res_{m_m}} \right) \end{pmatrix}$$

$$(15)$$

Equation (15) satisfies the requirements used for assigning resources to user request via EETACP (traffic load on DCCN server). From (13), if $V_{n_{n*M_m}} = 1$, then resources R_{m_m} will be precisely provide the tasks or job requirements T_{n_n} while maintaining the QoS requirements, but if $V_{n_{n*M_m}} < 1$, then the resource will be degraded by the tasks or job requirements T_{n_n} and affect server performance.

Again, in the DCCN, since user requests or demand arrives randomly into the gateway before the ISOLB, it was assumed that the job arrival follows the stochastic process such that the packet size is exponentially distributed, and the system is considered as an M/M/1 queuing system. Besides, traffic capacity management and optimum utilization of server resources takes care of network instability issues. But, to address the issue of traffic flow from the access layer, Little's

law suffices. It takes care of the system response time and scheduling distribution thereby maintaining traffic flow stability criterion.

Now, if the average arrival rate per unit time is given by $R_{L_{m*K}}$ and $P_{Vm_{L_m}}$ is the average service rate per unit time, stability condition for the system resource management can be deduced.

Considering the user task/job numbers (n_n), resource numbers (m_m) and QoS parameters (K_k), the proposed DCCN for enterprise cloud datacenter is considered stable only if $R_{L_{m*K}} < P_{Vm_{L_m}}$. If on the other hand, the average arrivals happen faster than the job service completions such that $R_{L_{m*K}} > P_{Vm_{L_m}}$, the job queues on the server clusters will grow indefinitely long and making the system to be unstable.

In this work, by generating an equivalent DCCN that will support the deployment of the proposed EETACP while exploring the mathematical models derived for the system, the QoS metrics evaluation will primarily be service availability, throughput, and fault tolerance. The algorithms for the distributed cloud management architecture were discussed previously. Fig 3 shows the testbed used for carrying out basic experimentations and validations in the context of application hosting and provisioning. Fig 4 shows the UNN datacenter which has network failures resulting from traffic imbalance and disturbances. This was addressed in validation section.

Fig. 3. Completed Testbed Setup for DCCN *Vm* Scheduling for Application Provisioning (SGEMS EEATCP).

Fig. 4. UNN Testbed Setup without *Vm* Scheduling for Application Provisioning

IV. SYSTEM VALIDATION

As explained previously, procedural benchmarking with server virtualization for the spine leaf DCCN was used in this work. This is driven by virtual machines and task schedulers. In carrying out the validation analysis on DCCN, procedural benchmarking was applied in creating the relevant heuristic algorithms. Virtual machines and task schedulers were configured as extended attributes after importing the objects from the object palette in Riverbed Modeller/C++ Version 17.5 simulator [60]. Other components configured include the OpenFlow load balancer and the server cluster links. The mathematical characterization above was considered in the design. After comparing the service delay and availability of the two DCCN using the heuristic algorithms for task length and user priority, this work then introduced two related datacenter architectures having task based scheduling without virtualization ie. DCell [5] and BCube [7]. Both have been extensively studied in [61]. The metrics computed includes average throughput (bytes/Sec), Average resource availability response and DCCN density on resource utilization. By extending the work carried out in [62] that focused on the impact of virtualization on server computing clusters, the contribution of the work now focused on server consolidation via virtualization for fault tolerance improvement in order to reduce down time scenario to the barest minimum. In this work, we focused on resource management using virtual machine for improved QoS. The simulation is done under the following conditions as enlisted in Table I, II and III.

TABLE I. BASIC CONFIGURATION

Number of Datacenter hosts	5
Number of Datacenters	5
Number of Brokers	1
Task Schedulers	10
Number of Cloudlets	10

Each datacenter consists of several hosts. Each host has its own configuration. Here, the same configuration is applied for each hosts. Host configuration is depicted in Table II.

TABLE II.   HOST BASIC CONFIGURATION

Buffer Capacity	256000
Storage	1GB
Bandwidth (MB/Sec)	10k
Number of Virtual Machines	6
MIPS (Lines of Codes)	1000

The host in the datacenter consists of several virtual machines. Each virtual machine has its own configuration. Here, same configuration is applied for each VMs. Virtual machine configuration is mentioned in Table III.

TABLE III.   VIRTUAL MACHINE BASIC CONFIGURATION

Number of Cores	2
Size (MB)	10000
Bandwidth (MB/Sec)	1000
RAM (MB)	512
MIPS (Lines of Codes)	1000

From Fig 5, it was observed that the proposed DCCN with optimal Virtual instance allocation coordinator had relatively a better throughput. The average throughput responses in percentages were obtained as follows: For DCCN = 40.00%, DCell = 33.33% and BCube = 26.67%. This shows that virtualization enhances performance.
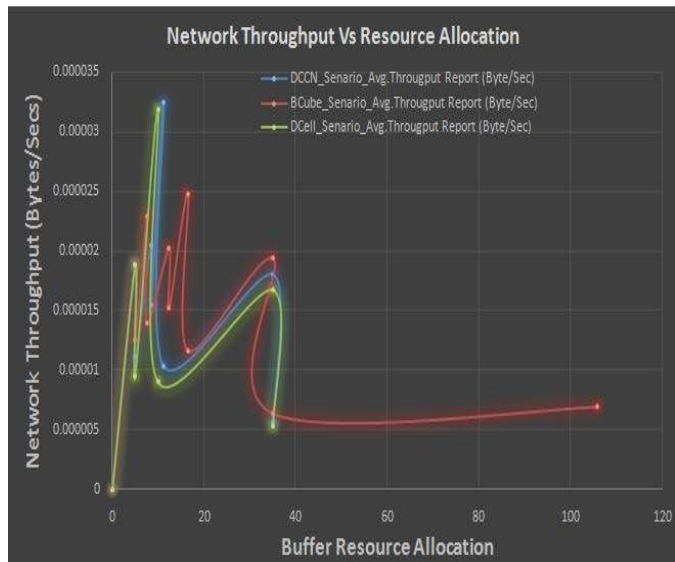

Fig. 5.   Plot of Network throughput against Resource Allocation

As shown in Fig.6, resource availability refers to ability to access the DCCN server clusters on demand while completing the job requests. It was shown that the proposed model had fairly good resource availability leading to enhanced performance. Hence, a good resource allocation strategy will enhance performance at large. This still re-echoes the advocacy for Vm based cloud networks particularly in the cell based and spine leaf models like the DCCN in this research.
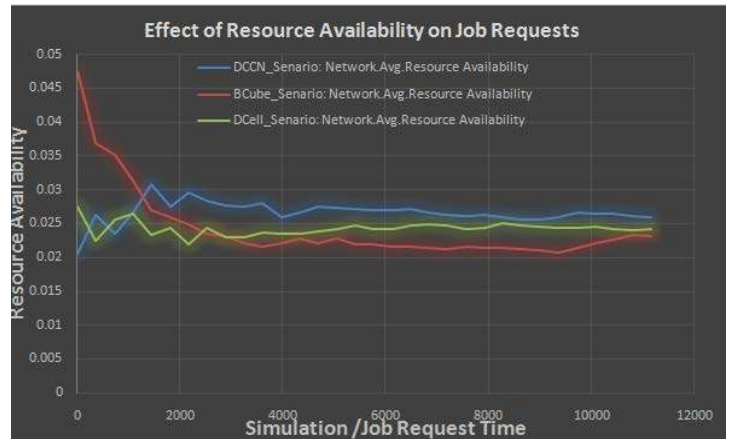

Fig. 6.   Plot of resource Availability Vs Job requests time

The legacy UNN DCN in Fig 4 can be improved with this scheme.

From Fig. 7, it was shown that lower resource utilization for the proposed DCCN was achieved compared with BCube and DCell scenarios. When all existing resources (VMs) are allocated to low priority jobs and a high priority job comes in, the low priority job (deadline is high) has to be pre-empted so that its resources can allow a high priority job (deadline is low) to run in its resource tasks. When a job arrives, availability of the VM is checked based on the network density. If the Vm is available, then job is allowed to run on the VM. If the VM is not available, then the algorithm find a low priority job taking into account the job's lease type. The low priority job pauses its execution by pre-empting its resource. In all cases, the high priority job is allowed to run on the resources pre-empted from the low priority. When any other job running on server VMs is completed, the job which was halted previously can be resumed. This *Vm* process in Algorithm1 facilitates lower resource utilization at large. DCCN density on resource utilization gave 40% (ie. when logically isolated) while the others offered 60% (ie. when not logically isolated).
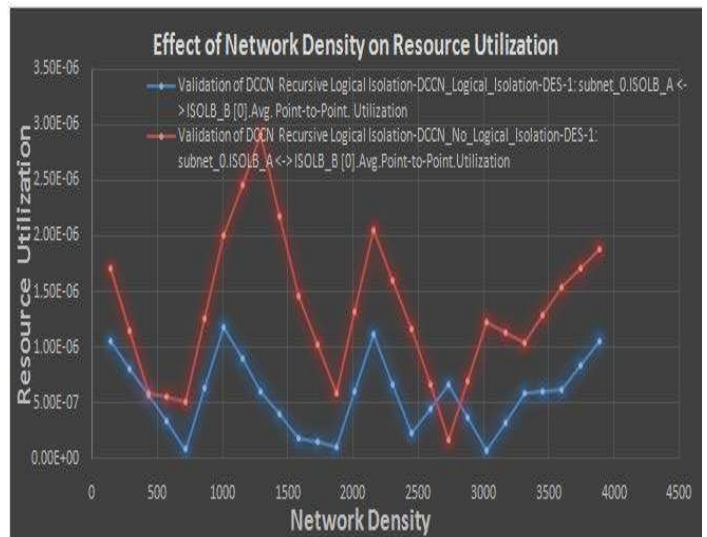

Fig. 7.   Plot of Network density Vs Resource utilization

V. CONCLUSION

This paper focused on the critical design factors involved in the deployment of SGEMS DCCN as well as analyzing the impact of virtualization on resource allocation and scheduling strategies for managing Cloud DataCenter server clusters. In this respect, optimizing computational resources (ie. network resources) with low cost are the vital considerations for a successful cloud computing service deployment and its operations. The work used CloudSim equivalent tool Riverbed 17.5 with scenario based setups for DCCN, DCell and BCube. The implemented network was simulated while ensuring that *Vms* were allocated as hosts based on the capacity of the cloud service coordinator available.

Jobs are given to the *Vms* for execution based on First Come First Serve (FIFO) basis. The deadline was checked for high and low priorities. From the work, QoS metrics such as throughput, resource availability, and resource utilization were investigated. With the latest state of art in enterprise application network, the proposed Spine leaf DCCN offered throughout improvement of 6.67% over DCell while offering 13.33% over BCube. Also, the network offered 5.13% availability improvement over DCell while offering 10.25% over BCube. Consequently, the more resource pool and allocation on a virtualized cloud server, the better the overall system performance owing to virtualization effects. This can also stabilize the DCN operations marginally. Conversely, an inefficient resource allocation/scheduling scheme can adversely degrade the network performance. In conclusion, virtualization can smoothly facilitate resource allocation/scheduling schemes in a distributed server domain considering user job workloads. Future work will focus on *Vm* allocation in high density Spine and leaf architecture for the deployment of EETACP service in SGEMS (microgrid framework) while building datasets for big data analytics using an FPGA based big data environment.

ACKNOWLEDGMENT

REFERENCES

[1] W. J. Dally and B. Towles. Principles and Practices of Interconnection Networks. Morgan Kaufmann Publishers, **2004.**

[2] M. A. Fares, A. Loukissas, and A. Vahdat, "A scalable, commodity datacenter network architecture," in Proc. SIGCOMM, Seattle, Washington, USA, **2008.**

[3] M. Kodialam, T. V. Lakshman, and S. Sengupta. Efficient and Robust Routing of Highly Variable Traffic. In *HotNets*, **2004**

[4] A. Greenberg, P. Lahiri, D. A. Maltz, P. Patel, and S. Sengupta. Towards a next generation data center architecture: scalability and commoditization. In PRESTO '08: Proceedings of the ACM workshop on Programmable routers for extensible services of tomorrow, pages 57–62, New York, NY, USA, **2008.** ACM.

[5] Chuanxiong Guo, Haitao Wu, Kun Tan, Lei Shi, Yongguang Zhang, Songwu Lu, "DCell: A Scalable and Fault-Tolerant Network Structure for Data Centers", in Proc. of SIGGCOM Seattle, Washington, USA, Aug.17–22nd , **2008.**

[6] H. Wu, G. Lu, D. Li, C. Guo, and Y. Zhang. "MDcube: a high performance network structure for modular data center interconnection", *In Proc. of the 5th international conference on Emerging networking experiments and technologies,(in CoNEXT '09),* Pp. 25–36, New York, NY, USA, **2009**. ACM.

[7] C. Guo, G. Lu, D. Li, H.Wu, X. Zhang, Y. Shi, C. Tian, Y. Zhang, and S. Lu. "BCube: a high performance, server-centric network architecture for modular data centers", *In Proc. of the ACM SIGCOMM 2009 Conference on Data communication, (in SIGCOMM '09),* Pp.63–74, New York, NY, USA, **2009**. ACM.

[8] Dan Li, Chuanxiong Guo, Haitao Wu, Kun Tan, Yongguang Zhang, Songwu Lu, "FiConn: Using Backup Port for Server Interconnection in Data Centers", *In Proceedings of INFOCOM'* 09, **2009.**

[9] Yong Liao ,Jiangtao Yin, Dong Yin , Lixin Gao, "DPillar: Dual-port server interconnection network for large scale data centers", *Elsevier Computer Networks* 56 (**2012**) 2132–2147.

[10] Udeze C.C, "Re-Engineering DataCenter Networks for Efficient Web Application Integration in Enterprise Organisations", PhD thesis, Unizik, February, **2013.**

[11] Okafor K.C and T.A Nwaodo: "A Synthesis VLAN Approach to Congestion Management in Datacenter internet Networks", In *International Journal of Electronics and Telecommunication System Research*, (IJETSR), Vol.5, No.6, May **2012**, Pp.86-92,

[12] LászlóGyarmati, Tuan Anh Trinh*", "*Scafida: A Scale-Free Network Inspired Data Center Architecture", In*ACM SIGCOMM Computer Communication Review*, Vol.40, No 5, October **2010**, Pp.5-12.

[13] Mosharaf Chowdhury, "Resource Management in Multi-Clusters:Cloud Provisioning", April 30, **2010**, www. aws.amazon.com/ec2.

[14] Abhishek Chandra, Weibo Gong, and Prashant Shenoy,"Dynamic Resource Allocation for Shared DataCenters Using Online Measurements",

[15] C.C.Okezie, Udeze C.C, Okafor K.C, "Cloud Computing: A Cost-Effective Approach to Enterprise Web Application Implementation (A case for web ERP model)", *In Academic Research International*, Volume 3, Number 1, July **2012**, pp 432-443.

[16] K.C, Okafor, C.C, Udeze. C, H. C. Inyiama, C.C Okezie, "Virtualization: A Cost Effective Approach To Infrastructure Management In Data Center Networks- (A Case for Elastic Sky X Server)", In *J. Basic. Appl. Sci. Res.*, 2(5), Pp.4910-4920, **2012**

[17] A. Bala and I. Chana, "Fault Tolerance-Challenges, Techniques and Implementation in Cloud Computing", In *International Journal of Computer Science Issues*, Vol. 9 Issue 1, No 1, Pp.288-293, **2012.**

[18] Whitepaper Vmware- Virtualization: Architectural Considerations and Other Evaluation criteria, [online]. Available: http://www.vmware.com.

[19] R. Mallavarapu, "Dynamic Resource Provisioning in IaaS Cloud Environment", Master's Thesis, Aalto Univers Espoo, August 9, **2012.**

[20] M.A. Sharkh, M.J., A. Shami, and A. Ouda, "Resource Allocation in a Network-Based Cloud Computing Environment: Design Challenges", *In IEEE Communications Magazine*, Pages 46-52, Nov. **2013.**

[21] . T. Tannenbaum, D. Wright, K. Miller, and M. Livny. Condor: A Distributed job scheduler. *Beowulf cluster computing with Linux*. MIT Press, Cambridge, MA, USA 307-350, **2001.**

[22] Platform Computing. *LSF Version 4.1 Administrator's Guide*, **2001**.

[23] http://www.platform.com/services/support/.

[24] R. Buyya, S.K.Garg, and R.N. Calheiros, "SLA-Oriented Resource Provisioning for Cloud Computing: Challenges, Architecture, and Solutions", In *Proc. International Conference on Cloud and Service Computing, IEEE 2011* .

[25] D.Warneke and Odej Kao, "Exploiting Dynamic Resource Allocation for Efficient Parallel Data Processing in the Cloud", In *IEEE Transactions On Parallel And Distributed Systems*, Pages 1-14, January **2011.**

[26] S. Kuribayashi, "Optimal Joint Multiple Resource Allocation Method for Cloud Computing Environments", In *International Journal of Research and Reviews in Computer Science (IJRRCS)*, Vol. 2, No. 1, Pages 1-8, March **2011**

[27] M.N. Bennani and D.A. Menasc´e, "Resource Allocation for Autonomic Data Centers using Analytic Performance Models", In Proc.

IEEE Int'l .Conf. on Autonomic Computing, Seattle, Washington, June13-16, **2005**.

[28] Vignesh V, Sendhil Kumar KS, Jaisankar N, Resource management and Scheduling in Cloud Environment, International Journal of Scientific and Research Publications, Volume 3, Issue 6, Pages 1-6, June **2013**

[29] FeiTeng. "Resource Allocation and Scheduling Models for Cloud Computing", M.Sc Thesis, Other. EcoleCentrale Paris, **2011**. English. <NNT : 2011ECAP0043>. <tel-00659303>

[30] Beloglazov , J. Abawajy, R.Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for Cloud computing" In *Elsevier Future Generation Computer Systems* 28 **(2012),** Pages.755–768.

[31] P.Devi, T.Gaba, "Implementation of Cloud Computing By Using Short Job Scheduling ", In *International Journal of Advanced Research In Computer Science and Software Engineering*, *IJARCSSE,* Volume 3, Issue 7, July **2013**, Pages 178-138.

[32] R.S. Indu "Efficient Job Scheduling Of Virtual Machines In Cloud Computing. *International Journal of Advanced Research in Computer and Communication Engineering* .Vol. 2, Issue 9, September **2013***, Pp. 3349-3354.

[33] M. Eisa, E. I. Esedimy , M. Z. Rashad , "Enhancing Cloud Computing Scheduling based on Queuing Models"*International Journal of Computer Applications (0975 – 8887)* Vol.85, No 2,Jan. **2014**.*Pp.17-23*.

[34] B. Rimal, E. Choi, I. Lumb, "A taxonomy and survey of cloud computing systems," In Proc. *IEEE Fifth International Joint Conference on INC, IMS and IDC*, pp.44–51, **2009**.

[35] M. Randles, D. Lamb, A. T. Bendiab, "A Comparative Study into Distributed Load Balancing Algorithms for Cloud Computing", In *Proc. IEEE 24th International Conference on Advanced Information Networking and Applications Workshops*, Pp. 551-556, **2010**.

[36] T. Gopalakrishnan Nair, M. Vaidehi, K. Rashmi, V. Suma, "An Enhanced Scheduling Strategy to Accelerate the Business performance of the Cloud System". In *Proc. InConINDIA 2012*, AISC 132, pp. 461-468, © Springer-Verlag Berlin Heidelberg **2012**.

[37] R.V.denBossche, K.Vanmechelen and J.Broeckhove, "Cost-Optimal Scheduling in Hybrid IaaS Clouds for Deadline Constrained Workloads", In *IEEEComp.Society 3rd International Conference on Cloud Computing,***2010.** DOI 10.1109/CLOUD.2010.58.

[38] S.Sawant, "A Genetic Algorithm Scheduling Approach for Virtual Machine Resources in a Cloud Computing Environment", M.Sc Thesis Project at San Jose State University, Online- http://scholarworks.sjsu.edu/etd_projects, **2011.**

[39] Steven S. Skiena, "The Algorithm Design Manual, 2nd Edition, Springer, **2008**, DOI: 10.1007/978-1-84800-070-4, ISBN: 978-1-84800-069-8 e-ISBN: 978-1-84800-070-4.

[40] A.Thomasa, Krishnalal G, Jagathy Raj V.P, "Credit Based Scheduling Algorithm in Cloud Computing Environment", In Proc.International Conference on Information and Communication Technologies (ICICT 2014),and Elseiver Procedia Computer Science 46, SciVerse ScienceDirect, Elsevier Inc.(2015) 913 – 920.

[41] Zhi Yang, Changqin Yin, Yan Liu. A Cost-based Resource Scheduling Paradigm in Cloud Computing. 2011-12th International Conference on Parallel and Distributed Computing. Applications and Technologies.

[42] Liang Luo, WenjunWu,Dichen Di, Fei Zhang, Yizhou Yan, Yaokuan Mao. A Resource Scheduling Algorithm of Cloud Computing based on Energy Efficient Optimization Methods.

[43] Qi cao,Zhi-boWei,Wen-mao Gong. An Optimized Algorithm for Task Scheduling Based On Activity Based Costing in Cloud Computing.

[44] J Blythe, S Jain, E Deelman, Y Gil, K Vahi. Task scheduling strategies for workflow-based applications in grids. Cluster Computing and the Grid; 2005.

[45] Bo Yang, Xiaofei Xu, Feng Tan, Dong Ho Park. An Utility-Based Job Scheduling Algorithm for Cloud Computing Considering Reliability Factor; 2011 International Conference on Cloud and Service Computing.

[46] Marcos D. Assuncao, Marco A. S. Netto, Fernando Koch, Silvia Bianchi. Context-aware Job Scheduling for Cloud Computing Environments. 2012 IEEE/ACM Fifth International Conference on Utility and Cloud Computing.

[47] Hsin-Yu Shih, Jhih-Jia Huang, Jenq-ShiouLeu. Dynamic Slot-based Task Scheduling Based on Node Workload in a MapReduce Computation Model.

[48] K. Kambatla, A. Pathak, and H. Pucha. Towards optimizing Hadoop provisioning in the cloud in USENIX Workshop on Hot Topics in Cloud Computing (HotCloud09); 2009.

[49] AtulVikasLakraa, Dharmendra Kumar Yadav, "Multi-Objective Tasks Scheduling Algorithm for Cloud Computing Throughput Optimization", "International Conference on Intelligent Computing, Communication & Convergence ICCC-2015, and Procedia Computer Science, SciVerse ScienceDirect, Elsevier Inc 48 (2015) 107 – 113.

[50] Ashish Kumar Singh,SandeepSahu, " Environment Conscious Public Cloud Scheduling Algorithm with Load Balancing", International Journal of Computer Applications (0975 – 8887) Volume 87 – No.13, February 2014, Pp.24-27.

[51] J. Octavio, Gutierrez-Garcia, Kwang Mong Sim, "A family of heuristics for agent-based elastic

Cloud bag-of-tasksconcurrent scheduling", Future Generation Computer Systems, SciVerse ScienceDirect, Elsevier Inc doi:10.1016/j.future.2012.01.005.

[52] DajiErgu, Gang Kou ,Yi Peng ,Yong Shi, Yu Shi, "The analytic hierarchy process: task scheduling and resource allocation in cloud computing environment", Springer Science+Business Media, LLC 2011

[53] Hongbo Liu, Ajith Abraham, Vaclav Snanel, SeánMcLoone, "Swarm scheduling approaches for work-flow applications with security constraints in distributed data-intensive computing environments", Information Sciences 192 (2012) 228–243.

[54] Young Choon Lee, Chen Wang, Albert Y. Zomaya, Bing BingZhoua, "Profit-driven scheduling for cloud services with data access awareness", J. Parallel Distrib. Comput. 72 (2012) 591–602 ScienceDirect.

[55] Wei Wang, Guosun Zeng, Daizhong Tang, Jing Yao, "Cloud-DLS: Dynamic trusted scheduling for Cloud computing", SciVerse ScienceDirect, Expert Systems with Applications 39 (2012) 2321–2329.

[56] XiaominZhua, ChuanHea, Kenli Li, Xiao Qin, "Adaptive energy-efficient scheduling for realtime tasks on DVS-enabled heterogeneous clusters", J. Parallel Distrib. Comput, SciVerse ScienceDirect, 2012 Elsevier Inc.

[57] Ye Huanga, Nik Bessis, Peter Norringto, Pierre Kuonend, Beat Hirsbrunner, "Exploring decentralized dynamic scheduling for grids and clouds using the community-aware scheduling algorithm", Future Generation Computer Systems, 2011 Elsevier doi:10.1016/j.future.2011.05.006.

[58] Vijindra, SudhirShenai, "Survey on Scheduling issues in Cloud computing, "International Conf. on Modelling Optimization and Computing ICMOC-2012) SciVerse Science Direct,Elsevier, Procedia Engineering 38 (2012)2881-288.

[59] Online: http://www.mathworks.com,. MATLAB Version 7.12.0.35 (R2011a)

[60] Riverbed Modeller- Available Online: https://supportstaging.riverbed.com/bin/support/static//doc/opnet/17.5.A/online/itguru_17.5.PL5/Models/wwhelp/wwhimpl/common/html/wwhelp.htm#href=Applications_Model_desc.02.05.html&single=true,

[61] K.C. Okafor, G.N.Ezeh, I.E.Achumba,O.U.Oparaku, U.Diala, "*DCCN: A Non-Recursively Built Data Center Architecture for Enterprise Energy Tracking Analytic Cloud Portal, (EEATCP), In AASCIT* Computational and Applied Mathematics Journal,*2015, USA. Pp.107-121.

[62] K.C Okafor, F.N. Ugwoke, O.U Oparaku, Characterization of Distributed Cloud Computing Network (DCCN) Server Resource Pool Using Virtualization Dynamics", *International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE) Vol 4, Issue 2, February 2015* 280-294.