

# Nutshell: Cloud Simulation and Current Trends

Ubaid ur Rahman, Owais Hakeem, Muhammad  
Raheem, Kashif Bilal  
Department of Computer Science  
COMSATS Institute of Information Technology  
Abbottabad, Pakistan  
{urahman, owaishakeem, mraheem,  
kashifbilal }@ciit.net.pk

Samee U Khan  
Department of Electrical and Computer Engineering  
North Dakota State University  
Fargo ND, USA  
samee.khan@ndsu.edu

Laurence T. Yang  
Department of Computer Science  
St. Francis Xavier University  
Antigonish, Canada  
ltyang@stfx.ca

**Abstract**—Cloud computing has experienced enormous popularity and adoption in many areas, such as research, medical, web, and e-commerce. Providers, like Amazon, Google, Microsoft, and Yahoo have deployed their cloud services for use. Cloud computing pay-as-you-go model, on demand scaling, and low maintenance cost has attracted many users. The widespread adoption of cloud paradigm upshots various challenges. The legacy data center and cloud architectures are unable to handle the escalating user demands. Therefore, new data center network architectures, policies, protocols and topologies are required. However, new solutions must be tested thoroughly, before deployment within a real production environment. As the experimentation and testing is infeasible in the production environment and real cloud setup, therefore, there is an indispensable need for simulation tools that provide ways to model and test applications, and estimate cost, performance, and energy consumption of services and application within cloud environment. Simulation tools providing cloud simulation environments currently are limited in terms of features and realistic cloud setups, focus on a particular problem domain, and require tool-specific modeling, which can be frustrating and time consuming. This paper aims to provide a detailed comparison of various cloud simulators, discuss various offered features, and highlight their strengths and limitations. Moreover, we also demonstrate our work on a new cloud simulator “Nutshell”, which offers realistic cloud environments and protocols. The Nutshell is designed to diminish flaws and limitations of available cloud simulators, by offering: (a) multiple datacenter network architectures, like three-tier, fat-tree, and dcell, (b) fine grained network details, (c) realistic cloud traffic patterns, (d) congestion control strategies and analysis, (e) energy consumption, (f) cost estimation, and (g) data center monitoring and analysis. Flexibility to stretch the architectures to simulate smart city IT infrastructure.

**Keywords**—cloud simulator; simulation; datacenter architectures; energy models; cloud computing; traffic patterns; datacenter congestion control; smart city

## I. INTRODUCTION

Cloud computing is an elastic platform, which aims to deliver computing resources and services on demand to users. Cloud computing is classified as (a) software, (b) platform and (c) infrastructure by services. Cloud is popular in research and business community for data analysis, research, simulations, e-commerce, and business applications. Cloud-based business services and Software-as-a-Service (SaaS) market is estimated to increase from \$13.4 to \$32.2 billion in period from 2011 to 2016 [1]. Similarly, growth of Infrastructure-as-a-Service (IaaS) and Platform-as-a-Service (PaaS) market is estimated from \$7.6 to \$35.5 billion in period from 2011 to 2016. Investments in cloud computing have delivered benefit yields around \$4 billion in the last five years.

Cloud computing core component is datacenter, which is a facility where computing resources are interconnected via communication infrastructure, used for storage and application hosting [2], [3]. Cloud computing is subjected to growth in terms of scale, users, and complexity. Therefore, ensuring stability, availability, and reliability is of vital significance, which mandates new scalable datacenter architectures, policies for scheduling, Service Level Agreement (SLA) for ensuring Quality of Service (QoS), protocols for datacenter network (DCN) to handle issues, such as congestion, routing and Virtual Machine (VM) migrations. New solutions for datacenters or cloud must be tested before their implementation in a real production cloud environment. Testing within a real data center environment is expensive and infeasible, e.g., a 20% revenue loss is reported by Google because of an experiment that caused an additional delay of 500ms in response time [1]. Therefore, simulation frameworks, which depict realistic cloud environments are the feasible alternatives to test and analyze

in detail the proposed architectures, protocols, and strategies.

There are number of simulation tools available for cloud computing, which provide a platform for developers and researchers. However, the available simulators are limited in terms of functionalities, and are unable to provide a realistic cloud environment for fine grained analysis and testing. Therefore, the simulation results achieved in an environment, which is unable to depict the realistic cloud behavior deviates from realistic scenarios and cannot be deployed in real cloud environments.

In this paper we present an analysis and comparison of the state-of-the-art cloud simulators, highlighting offered features and pointing major limitations in cloud simulators. We also highlight the significance and pivotal role of communication network infrastructure, network traffic patterns, and congestions control mechanisms proposed for cloud data centers. Based on the observed limitations in the existing cloud, and to offer realistic cloud environment for simulation, we outline the proposal for a new cloud simulator named Nutshell.

The rest of paper is organized as: Section II details current cloud computing simulators, analysis and comparison of existing cloud simulators. Section III explains various types of traffic patterns in datacenters and their need in modeling of cloud architecture. Section IV details the significance and role of congestion control mechanisms in cloud environment. Section V demonstrate current work on the “Nutshell” simulator. Section VI briefly introduces the use of Nutshell in smart city IT infrastructure. Section VII concludes the paper with summarized comparison of Nutshell with existing tools.

## II. CLOUD SIMULATORS

### A. *CloudSim*

CloudSim [4] built upon GridSim [5] is an event driven simulator. CloudSim provides simulation features for modeling and simulating datacenter environments, Virtual Machines (VM), brokering policies and pay-as-you-go cloud model. It is built in Java, providing different modeling components, and easy extensibility. Development in java is also a drawback, since in a 32 bit systems, Java can handle at most 2GB of memory [6].

Datacenter resources are viewed as a collection of VM by CloudSim [7]. CloudSim is faster and has the ability to scale into larger number of datacenter node [8]. It captures object interaction effect instead of building small simulation objects, reduces complexity but lacks simulation accuracy. CloudSim has limitation on infrastructure simulation [9]. CloudSim models Inter-connection between cloud entities based on conceptual networking abstraction.

There is no support for a datacenter network topology, only basic network model and limited workload generator is available. Directed graphs are used to maintain network topology [8], and network latency is modeled using a matrix

that specifies the delay between two simulated entities[9]. Information of network topology is stored in BRITE format, loaded every time and is used to build a latency matrix. Edges are assigned bandwidth and delays, during transmission, if an edge is used then for transmission delay duration, the bandwidth component is reduced [8]. The loaded information is used to add delay to a message [9]. The delay is not always the imitation of realistic practical delay in dynamic networks.

There is no control over network topology component’s configuration such as, routing protocols, datacenter internal organization, etc., affecting the overall throughput and performance. CloudSim cannot capture protocol dynamics (congestion control, error recovery or routing specifics) and unable to measure network overhead, resources or provisioning policies [8] [10]. There are no models for Network Interface Card (NIC) or links between components. The application model fits well with applications that are computationally intensive having no specific completion deadline, High Performance Computing (HPC).

There is no or limited implementation of Communication of data model in application model [7]. It has no implementation of Business Process Management (BPM) and Service Level Agreement (SLA) components. CloudSim do not consider energy consumption of datacenter resources.

### B. *NetworkCloudSim*

NetworkCloudSim [7] an extension of CloudSim, supports communication between the application element and various network elements. The data flow between VMs on host and across network is modeled to capture delay in transmissions. Application classes of CloudSim are extended in order to denote a generalized task with various stages, i.e., computation, sending or receiving data. To make application model aware of network, scheduling models are extended. NetworkCloudSim two levels of scheduling, a Host level (scheduling VMs on host) and VM level (execution of real applications).

It adds the functionalities of network topology and application modelling to CloudSim. There are three switch models (root, aggregate and edge) and can be configured as a router. It has the ability to model delays when forwarding data to other host, or switch. This enable modeling different network topologies.

### C. *CloudAnalyst*

CloudAnalyst [11] is also built with CloudSim at its core, it provides a graphical user interface (GUI) for ease of use. CloudAnalyst add models that features Internet and Internet Application behavior to CloudSim. Its basic purpose is to evaluate performance and cost of SaaS datacenters [12] distributed geographically, having heavy user load. It allow configuration of any geographical distribute system. It allow adding new service brokering policy to control the users of any geographical location

based on services provided by distributed datacenters. Experiment output includes request response time and cost of keeping the infrastructure, which is based on the cost policy of Amazon EC2 [12].

#### D. EMUSIM

EMUSIM provides both simulation (based on CloudSim) and emulation (based on automated emulation framework—AEF) of a cloud application. It is developed for SaaS, applications having huge CPU-intensive. Emulation use application behavior to extract information automatically, and use this to generate more accurate simulation models and help estimate performance and cost of a cloud application.

#### E. CDOSim

CDOSim [13] is a cloud deployment options (CDO) extending CloudSim. It can simulate CDO's response time, SLA violations, and costs. It provide abstraction to users from fine-grained details of cloud platform, and are available to providers. It provides a benchmark to test the impact of architecture of cloud platform on performance of an application. Application models comply with the knowledge Discovery Meta-Model (KDM) of OMG, any application model created with automated reverse engineering tools with the capability of creating a KDM, can be deployed for performance analysis. Production monitoring data can be used to create workload profiles.

#### F. MR-CloudSim

MR-CloudSim [14] is also an extension to CloudSim, providing an easier and economical way to interrogate MapReduce model. MapReduce is used in excess with big data, CloudSim do not support file processing, along with cost and time related with it, and thus MR-CloudSim extends CloudSim to add big data processing analysis technique. Datacenters these days store huge amount of information because of increase in data consumption and availability of high network bandwidth. MR-CloudSim is not tested with the existing industry approved, real MapReduce model Hadoop [33].

#### G. TeachCloud

TeachCloud [9] is another extension to CloudSim. It is a research-oriented simulator for development and validation use in cloud computing. It provides a platform for experimentation of various cloud platform components like, datacenters, datacenter networking, processing elements, virtualization, SLA constraints, BPM, Service Oriented Architecture (SOA), web-based applications, management and automation. New network topologies like DCell, VL2, Portland, and BCube are also part of the extension. TeachCloud provides a simple graphical interface (GUI) for cloud configuration and experimentation. Cloud workload generator is also added to CloudSim. A monitoring outlet is also introduced for most of the cloud platform components. Reconfiguration of cloud system is achieved with an action

module that is added to CloudSim, which allow evaluating the impact of changes on the system total performance.

#### H. DartCSim+

DartCSim+ [10] is also an enhancement to CloudSim. It integrated power and network models, also making network and scheduling algorithms power-aware. Models for network links and NICs also exist. It has extended the migration of virtual machines, by considering network overhead for accuracy of the process. There is also a mechanism for failed packets resubmission caused by either hardware failure or due to live migration of VM, policy implemented for the mechanism is customizable.

Mechanism for controlling transmission of network links is also added which solve the problem of distortion. Corresponding events are added to implement and control a basic first in first out (FIFO) network scheduling policy.

#### I. GreenCloud

GreenCloud [8] is a simulation tool, an extension of NS2 (a packet-level network simulator) with the purpose of analyzing and experimentation with energy-aware cloud computing datacenters. It can capture energy consumption, provide fine-grained details of datacenter components such as servers, switches and links, workload distribution and a realistic packet-level communication patterns providing finest-grained control. GreenCloud code is written in C++ with OTcl layer on top, which is a drawback since users have to learn both the programming language. It require minutes to simulate and a huge amount of memory, limiting the simulation to small datacenters only[7].

The user application models are implemented as simple objects, which describe computational requirements. Application's communicational requirements are indicated in terms of data amount to be transferred before and after the completion of a task. User application model is also extended to adding a predefined execution deadline to implement QoS requirements defined in SLA.

It allows the incorporation of different communication protocols like TCP, IP, UDP in a simulation [7].

The simulator lags the ability to find energy consumption in Storage Area Network (SAN). To minimize resources during job selection DENS [15] is used considering workload and communication capability of datacenter [16]. GreenCloud, experiments can only use one machine resources [6]. There is no model for virtualization in GreenCloud [17].

#### J. MDCSim

MDCSim [18] is a comprehensive, flexible, and scalable discrete event simulator, available commercially due to its core being CSim [19] which is a commercial product [7][6]. The whole simulation model is configured into three layers: a communication layer, which supports IBA and Ethernet over TCP/IP as interconnect technology. This layer also support major functionalities of the IBA. A kernel layer, where the Linux scheduler 2.6.x is modeled, and maintains a

run queue for each CPU in the system; and a user-level layer, which captures the vital characteristics of a three-tier architecture. Categories of processes in user level layer are: WS, AS, and DB processes; communication processes like Sender/Receiver processes; auxiliary processes like disk helper for complementing server processes.

Specific hardware features of different datacenter components like servers, switches, and communication links collected from various vendors can also be modeled in MDCCSim, allowing estimation of power consumption [7] [8]. Only applications with computation tasks are supported. The effect of object interaction is captured instead of building and processing small simulation objects (like packets) individually [8], therefore it lacks simulation accuracy.

Its communication model works same as CloudSim. MDCCSim do not capture any protocol dynamics for congestion control, routing specifics, or error recovery. MDCCSim performs only rough estimation on power consumption only for computing servers, and do not monitor communication-related, for a given monitoring period, it uses special heuristics averaging on the number of the received requests. User must have the knowledge of C++/Java in order to use this simulator. MDCCSim experiments can only use resources of single machine [6]. MDCCSim do not consider virtualization and multiple tenants of the datacenter [17].

#### K. *iCanCloud*

*iCanCloud* [6] is a software simulation framework for large storage networks of cloud computing architectures, developed based on SIMCAN. It provides customizable global hypervisor, for implementation of any brokering policies; the simulation framework include Amazon's provided instance types, for comparison with a corporate model. VMs can be modified to simulate uni-core/multi-core systems. *iCanCloud* has been designed to perform simulation on several machines in parallel. It is written in C++, allowing use of all memory available on experiment running machines. It can create accurate simulation experiments with detailed physical layer entities simulation like cache, memory allocation policies and file system models, there is no power consumption model in *iCanCloud*.

A wide range of storage system models (NFS, RAID systems, and parallel file systems) are included in *iCanCloud*. *iCanCloud* has a user-friendly GUI.

For modeling and simulating applications *iCanCloud* provides a POSIX-based API and an adapted MPI library. New components can be added in order to increase the functionality of the simulation platform. *iCanCloud* has the ability to predict trade-off between an application's costs and performance on a specific hardware to notify user about the involved cost. It focuses on policies, which charge users in a pay-as-you-go manner. Inter-cloud functionality is not available in *iCanCloud* [20].

#### L. *GroudSim*

*GroudSim* [21] is an event-based simulator developed for scientific applications explicitly, running on Grid and Cloud environments, and need only one simulation thread. It is flexible, concerned with IaaS service, but can easily be extended to support additional models such as PaaS, DaaS and TaaS. *GroudSim* has implementation of time advance algorithm, clock, and *future event list* (FEL). Indirection level is added for forwarding events directly to the destination entity, to allow manipulation of entities state. It implements resource policy sharing in case of sharing Cloud resources.

*GroudSim* supports two cost models, use base model for cloud and CPU core based for Grids. It allows keeping track of cost results, and supports custom billing intervals to study their effect on overall cost.

*GroudSim* has implementation of two configurable tracing, entity state tracing and event-based tracing. *GroudSim* base programming language is Java. *GroudSim* can easily be extended by adopting probability distribution packages. Upon failure, user can change the definition of error behavior in *GroudSim*.

#### M. *DCSim*

*DCSim* [17] is an event-driven simulator focused on a virtualized datacenter providing IaaS to any multiple tenants, designed to provide an easy framework for developing and experimenting with datacenter management techniques and algorithms. It focuses on modelling transactional, continuous workloads (such as a web server), but can be extended to model other workloads (such as application workload) as well. *DCSim* provides the additional capability of modelling replicated VMs sharing incoming workload as well as dependencies between VMs that are part of a multi-tiered application. SLA achievement can also be more directly and easily measured and available to management elements within the simulation. *DCSim* (Data Centre Simulator) is an extensible data center simulator implemented in Java.

It provide extensible and customizable points to insert new management algorithms and techniques.

It has multiple interconnected hosts and each host having own CPU scheduler and resource managing policy. *DCSim* simulates datacenter with the centralized management system. It neglects datacenter network topology for higher scalability.

#### N. *SimIC*

Simulating the Inter-Cloud (*SimIC*) [20] is a discrete event simulation toolkit based on *SimJava*, mimicking inter-cloud activity. It includes the fundamental entities of the inter-cloud meta-scheduling algorithm such as users, meta-brokers, local-brokers, datacenters, hosts, hypervisors and virtual machines (VMs). It also provide additional features of resource discovery and scheduling policies along with VMs allocation, re-scheduling and VM migration strategies.

SimIC accepts optimization of different selected performance criteria for entities diversity.

It allows reactive orchestration based on current workload of already executed heterogeneous user specifications, present in the form of text files that can be load in real-time at different simulation intervals. It is capable of performing service contracts (SLAs). ICMS algorithm is used by SimIC for inter-cloud scheduling depending upon most of the distributed parameters. It also provides pay-as-you go model.

It does not have energy model reduction of power consumption during message distribution, host scheduling policy for time sharing.

#### O. SPECI

Simulation Program for Elastic Cloud Infrastructures (SPECI) [22] is a simulation tool, which allows exploration of aspects of scaling as well as performance properties of future DCs. Basic purpose of SPECI is to evaluate the performance and behavior of datacenters, with size and middleware design policy provided as input. The existing Java package DES is used in SPECI. It has two packages one represent datacenter topology and other contains the components for experiment execution and measuring. Event scheduling as well as random distribution drawing are implemented by SimKit, it also enable modeler to execute repeated runs with different configurations. Statistical analysis of the output is handled by the simulation entry class.

It has implementation of models for different component in the datacenter, such as nodes and network links, mimicking observed datacenter network operations. The components have monitoring points that can be activated as required by the experiment.

The network topology is assumed to be a one hope switch, and it does not implement any routing and network workload.

SPECI maintains a global view of the datacenter to deal with the connection and referral logic, used only with centralized heartbeat retrieval or policy distribution topology, such as the central or hierarchical one.

*Table 1 and 2 summarizes existing cloud simulators.*

### III. DATACENTER TRAFFIC PATTERNS

The performance of DCN and other distributed systems depends on infrastructure, protocols, and real time traffic patterns. Traffic pattern is flow of data inside a datacenter and flow to and from the Internet. The flow within the data center can be one to one, one to many or many to many. Real time traffic is highly unpredictable due to system failures, elephant and mice flows, and tradeoff between energy efficiency and performance of a system. To have a better understanding of datacenter traffic flow, samples need to be taken with respect to time and usage in deferent scenarios. It is also worthwhile to use reserved features of IP and TCP to monitor and control patterns on a computing

system [23]. Better understanding of DCN traffic can be accomplished through measurement and analysis of real time traffic, which is resource, energy hungry and a time consuming effort. The resulting comprehensive view is a bit difficult to analyze but useful for the future development. The most important part is to know the traffic flows, extract information of the patterns, which is quite challenging, since datacenter traffic traces are not available publically, due to privacy and security issues [24]. However some researchers examined traffic patterns of real Data centers. Kandula et.al [25] collected events from 1500 servers for over two months. They examined that, probability of 89% of no traffic exchange among the servers within the same rack and 99.5% for the pairs in different racks. Either a server exchange traffic with all the servers within the rack or to fewer than 25% may doesn't communicate to the servers outside its rack, or it only about 1-10% of the rest, outside servers. They examine that some links are utilized highly, 86% of links are still observed congested at least for 10 seconds and 15% observed congestion lasting for 100 seconds at least. The congestion most of the time tends to short leaved, i.e., over 90% of them were in 2 seconds limit, whereas long lasting congestion exist.

In a single day of monitoring there were 665 unique scenarios of congestion, each of them lasted for at least 10s few of them lasted even for several hundreds of seconds and the longest of them were recorded 382 seconds. This above nature was actually recorded among 150 switch links.

Taking flow in consideration, traffic changes very frequently. There were few long flows in which .1% cross the 200s figure. Centralized decision in terms of choosing which path a certain flow should take is challenging, as in DC more than 50% bytes are in the flows which last no longer than 25 seconds.

Though many techniques have been proposed recently, which Benson et.al discloses that those techniques can only achieve 80% up to 85% of the ideal solution in terms of the number of delivered bytes [26]. The vital and more exhausting work is to provide traffic patterns environment so that to reduce congestion at high aggregate links and ensure smooth flow among them. Since we have sudden spikes in traffic which makes it important to handle them by providing patterns [27]. If we fail to provide traffic engineering algorithm it may overload network links and routers unnecessarily, cause long delays, high rate packet loss, reduced network throughput (e.g., TCP flows), which reduces the network reliability and efficiency, hence lead to SLA violation, and results in potential financial losses.

Benson et.al [28] selected 19 different data centers and examined their collected SNMP data. They carried out an in depth study of time based and spatial fluctuations in traffic size and packet loss in data center network. They found that core switches were highly loaded than edge switches but the loss rate was contrary to traffic load. The loss rate was high at edges and lower at core switches. Moreover they

**Table 1 Summary of analysis of CloudSim and its extensions**

		CloudSim and Extensions							
		CloudSim	Network CloudSim	Cloud Analyst	EMUSIM	CDOSim	MR-CloudSim	TeachCloud	DartCSim+
Parameters	Platform	SimJava	CloudSim		CloudSim, AEF	CloudSim			
	Language	Java							
	Availability	Open Source				-	Open source		
	Graphical User Interface	x	x	✓	x	✓	x	✓	x
	Application Model	✓	✓	✓	✓	✓	✓	✓	✓
	Communication Model	Limited	Full	Limited	Limited	Limited	Limited	Limited	Full
	Energy Model	✓	✓	✓	✓	✓	✓	✓	✓
	VM Support	✓	✓	✓	✓	✓	✓	✓	✓
	SLA Support	x	x	x	x	✓	x	✓	x
	Cost Model	✓	✓	✓	✓	✓	✓	✓	✓
	Network Topology Model	x	✓	x	x	x	x	✓	✓
	Addressing Schemes	x	x	x	x	x	x	x	x
	Congestion Control	x	x	x	x	x	x	x	x
	Traffic patterns	x	x	x	x	x	x	x	x

**Table 2 Summary of analysis of Cloud simulators**

		Cloud Simulators						
		GreenCloud	MDCSim	iCanCloud	GroudSim	DCSim	SimIC	SPECI
Parameters	Platform	NS-2	CSim	OMNET, MPI (SIMCAN)	-	-	SimJava	Java DES
	Language	C++, OTcl	Java, C++	C++	Java	Java	Java	Java
	Availability	Open Source	Commercial	Open Source	Open Source	Open Source	-	Open source
	Graphical User Interface	Limited	x	✓	Limited	x	x	x
	Application Model	✓	✓	✓	✓	✓	✓	✓
	Communication Model	Full	Limited	Full	x	x	Limited	x
	Energy Model	✓	Rough Estimation	x	x	x	Rough Estimation	x
	VM Support	x	x	✓	✓	✓	✓	✓
	SLA Support	✓	x	x	x	✓	✓	x
	Cost Model	x	x	✓	x	✓	✓	✓
	Network Topology Model	✓	Limited	✓	x	x	x	x
	Addressing Schemes	x	x	x	x	x	x	x
	Congestion Control	x	x	x	x	x	x	x
	Traffic patterns	x	x	x	x	x	x	x

found that about 40% of links remained idle but were constantly changing, making it hard to utilize the unused links. Hence the bursty and unpredictable nature of data center traffic makes the present traffic engineering schemes less appropriate.

The authors in [29] studied the efficiency of different traditional traffic engineering techniques (normally ECMP based) and experienced that these schemes are insufficient for data center network and provided three different reasons: (1) absence of multipath routing, (2) absence of load understanding and (3) absence of an overall view traffic engineering decision making. They have studied the data center's traffic patterns very closely and observed that there is short term predictability in data centers with duration which only last for few seconds. They have also suggested requirements for a good traffic engineering technique such as utilization of path diversities, coordination in traffic scheduling and adaptation of short term predictability and at the end they have presented their own frame work MicroTE. This leads to the conclusion that having traffic patterns in simulators is of vital importance in order to study DCN behavior.

Simulators discussed in section II failed to offer this feature. Our current work on cloud simulator Nutshell aims to overcome this shortcoming of existing simulators. Based on the knowledge of datacenter traffic, we aim to provide application models that depicts the traffic patterns of a real datacenter. The application models for datacenter traffic patterns, will result an accurate cloud simulation, providing a realistic view of datacenter and accurate calculation of delay, bandwidth, energy, performance and cost.

#### IV. DATACENTER NETWORK CONGESTION CONTROL

Datacenter network require proper consideration, since it is the backbone of cloud platform. Datacenter network is confronted with different challenges [28], [30], among which flow transmission and congestion control is a big issue. Increase in intra-datacenter communication is expected [31]. The flows in DCNs can be classified into short and long flow based on size, their simultaneous existence may lead to congestion resulting in performance degradation.

Working in a cloud environment, the service providers and developers need to anticipate the change in network traffic patterns and adjust swiftly to these changes or at least find system's bottlenecks. Hence, congestion control must be considered when working in cloud environment, and must be a part of simulator. Congestion control is not implemented in any simulator discussed in section II.

Many congestion control schemes for datacenter network are presented to root out the congestion problem. Among these schemes some are deadline aware schemes like D<sup>2</sup>TCP [32] and D<sup>3</sup> [33], and some are deadline unaware like DCTCP [34] which is TCP based. CONGA

[35], and Hedera [24] are multipath flow forwarding schemes. DCTCP [34] was presented to address tail latency. It shortened tail latency somehow but it was not deadline aware. D<sup>3</sup> [33] is first deadline aware flow scheduler which was presented to root out limitations present in DCTCP. D<sup>3</sup> shortened the tail latency and was also deadline aware. D<sup>3</sup> is centralized scheme which uses a proactive approach. For bandwidth allocation it uses first come first serve approach. It does not consider deadline parameter when allocating the bandwidth which is a big problem. Another practical shortcoming with D<sup>3</sup> is that it cannot coexist with TCP. To eliminate the shortcomings of D<sup>3</sup> another scheme D<sup>2</sup>TCP was proposed. D<sup>2</sup>TCP [32] is a deadline aware scheme which handle the fan-in bursts. It is a decentralized scheme which uses reactive approach in which sender react to the congestion. It back-off those flows which have far deadline and allocate bandwidth to those flows which have near-deadline. It requires no information about other flows. It is also compatible with legacy TCP.

CONGA [35] is a distributed multipath flow forwarding scheme which splits TCP flow into small flowlets and estimate real-time congestion on paths. CONGA uses a two party (leaf to leaf) mechanism to convey path wise congestion metrics between pairs of top of the rack switches in DC. CONGA uses global congestion information. The remote switches provide feedback and on the basis of which paths are assigned to the flowlets. This enables CONGA to efficiently balance the load without requiring any TCP modifications. CONGA achieves two to eight times better throughput than MPTCP [35]. Hedera [24] is a dynamic flow scheduling technique for multipath topologies of DCN. Hedera exploits the path diversity in DCNs topologies to enable near optimal bisection bandwidth for a range of traffic patterns. Hedera target long lived flows. The operation of Hedera includes three steps. First it detect long flows at edge switches. 2nd it estimates natural bandwidth demand of elephant flows and uses placement algorithm to compute path for the flow. Simulated annealing placement algorithm is used for placing the elephant flow and finally these paths are installed on switches.

Absence of such schemes for congestion control in current simulators, motivates our current work on Nutshell, to provide congestion control in simulation of cloud environment. This feature will facilitate researchers and modelers to experiment with a setup close to real network environment of cloud, allowing implementation of congestion based algorithms. Table 3 shows different characteristics of MPTCP, Hedera and CONGA, such as, balancing network load, targeted flows, and their implementation category. While table 4 shows different properties of DCTCP, D<sup>2</sup>TCP and D<sup>3</sup>, such as, deadline, flow priority, handling network traffic burst, and their compatibility with legacy TCP.

**Table 3 Comparison of MPTCP, Hedera, and CONGA**

Scheme	Control Plane	Method for network Load Balancing	Targeted Traffic flow	Implementation
MPTCP	Distributed	Several addresses from the same source and destination+ ECMP	All kind of flows	Modified TCP/IP stack
Hedera	Centralized	Hash-based dispatching+ routing of flow based on link utilization	Elephant flows	Openflows
CONGA	Distributed	Flowlet routing	All kind of flows	Custom switching ASICs

**Table 4 Comparison of DCTCP, D<sup>2</sup>TCP, and D<sup>3</sup>.**

Scheme	Deadline aware	Flow priority	Burst Tolerance	Compatibility with TCP
DCTCP	No	No	High	Yes
D <sup>2</sup> TCP	Yes	Yes	Low	Yes
D <sup>3</sup>	Yes	No	Medium	No

## V. STATE-OF-THE-ART CLOUD SIMULATOR : NUTSHELL

In this section we describe on going work on a new cloud simulator *Nutshell*, which is an extension to existing Network Simulator NS3, developed in C++. Current work involve developing complete module that encompasses datacenter network topologies, addressing schemes, customizable scheduling and policies, capturing energy consumption by components, congestion control, and traffic pattern. The main objective is to develop it as modular as possible, so that dependencies between module components is reduced, and allow other components to be plugged in with ease.

In cloud computing, datacenter is the most critical part. Performance, cost, and energy consumption are affected directly from its architecture. Hence, it is essential to provide models that reflect real world datacenter network in order to test applications with accuracy. Current development of simulator includes modeling of all necessary components that makes a datacenter network. The module contains network topologies such as: three-tier, fat-tree, and DCell, which are configurable i.e. it allows users to set the architecture component like, nodes processing capabilities, their bandwidths, switches, protocols running on those switches and their addressing schemes.

Another goal of implementing fully featured modules is to facilitate developers, modelers and providers; with the implementation of Nutshell, user can create complete datacenter network with a single configurable object. This will allow researcher, developers, modeler and providers to focus on the goal itself, instead of worrying about fine-grained details of datacenters.

Keeping in mind customization, the simulator is structured, so that any enhancement desired by researcher or user, can easily be plugged it, without interacting with existing code. This will enable user to plug in their developed policies, scheduling algorithms and protocols to study the behavior of application or network, its impact on cost, performance and energy consumption. Energy

consumption by different components in Nutshell is more realistic as compared to other simulators, due to the consideration of realistic datacenter network. The energy models gather energy consumption on each components. It also includes tracing simulation data for analysis of responses by datacenter, making it easy to focus on the part where actual customization might improve different aspects of datacenter network.

The core simulator provides a platform where different new components can be added in order to enhance or customize its ability to simulate a datacenter network for accurate results.

## VI. NUTSHELL AND SMART CITY

There are number of definitions of smart city [36], however we select a simple definition of a smart city, which is, A city “connecting the physical, IT, social, and business infrastructures to influence the city’s collective intelligence” [37]. A smart city relies on a collection of smart computing and other technologies, which are applied to critical services and infrastructure components. Smart computing can be defined as “a combination of new generation hardware, software, and network technologies helping in real-time awareness of IT systems, and people in order to make intelligent decisions regarding alternative actions resulting in optimized business processes.” [36].

Nutshell provides different architecture for datacenter network infrastructures, these architecture are designed as such that the infrastructure can adopt any IT infrastructure such as for smart city. Nutshell is subjected to provide configurable objects that can implement any user’s policies, protocols and schedulers, enabling a researcher to implement his/her work easily and focus on the general architecture building rather than worrying about minute details that entails complete architecture for smart city IT infrastructure.

## VII. CONCLUSION

From the discussion in this paper, it is evident that current simulators for cloud platform, only provide some features while lag behind in other, which are also necessary for assessing different aspects of a cloud platform. The



absence of real network model in most simulators, no network addressing schemes, no congestion control and real datacenter traffic patterns consideration in all of them, demands some attention. These requirements encouraged the work on *Nutshell*, an NS3 extension, and is currently under development. *Nutshell* aims to provide a realistic datacenter model for simulation, testing of policies and protocols, capturing cost, performance and energy details of datacenter.

#### REFERENCES

- [1] K. Bilal, O. Khalid, S. Ur, R. Malik, M. Usman, and S. Khan, "Fault Tolerance in the Cloud," no. ITPortal, pp. 1–13, 2012.
- [2] K. Bilal, "A Quantitative Comparison of the State of the Art Data Center Network Architectures," p. 6.
- [3] K. Bilal, S. U. R. Malik, O. Khalid, A. Hameed, E. Alvarez, V. Wijaysekara, R. Irfan, S. Shrestha, D. Dwivedy, M. Ali, U. Shahid Khan, A. Abbas, N. Jalil, and S. U. Khan, "A taxonomy and survey on Green Data Center Networks," *Futur. Gener. Comput. Syst.*, vol. 36, no. c, pp. 189–208, 2014.
- [4] R. N. Calheiros, R. Ranjan, C. A. F. De Rose, and R. Buyya, "CloudSim: A Novel Framework for Modeling and Simulation of Cloud Computing Infrastructures and Services," *arXiv Prepr. arXiv0903.2525*, p. 9, 2009.
- [5] R. Buyya and M. Murshed, "Gridsim: A toolkit for the modeling and simulation of distributed resource management and scheduling for grid computing," *Concurr. Comput. Pract. ....*, pp. 1–37, 2002.
- [6] A. Núñez, J. L. Vázquez-Poletti, A. C. Caminero, G. G. Castañé, J. Carretero, and I. M. Llorente, "ICanCloud: A Flexible and Scalable Cloud Infrastructure Simulator," *J. Grid Comput.*, vol. 10, pp. 185–209, 2012.
- [7] S. K. Garg and R. Buyya, "NetworkCloudSim: Modelling parallel applications in cloud simulations," in *Proceedings - 2011 4th IEEE International Conference on Utility and Cloud Computing, UCC 2011*, 2011, no. Vm, pp. 105–113.
- [8] D. Kliazovich, P. Bouvry, and S. U. Khan, "GreenCloud: A packet-level simulator of energy-aware cloud computing data centers," *J. Supercomput.*, vol. 62, no. 3, pp. 1263–1283, 2012.
- [9] Y. Jararweh, Z. Alshara, M. Jarrah, M. Kharbutli, and Alsaleh M.N., "TeachCloud: a cloud computing educational toolkit," *Int. J. Cloud Comput.*, vol. 2, no. 2012, pp. 237–257, 2012.
- [10] X. Li, X. Jiang, K. Ye, and P. Huang, "DartCSim+: Enhanced cloudsims with the power and network models integrated," in *IEEE International Conference on Cloud Computing, CLOUD*, 2013, pp. 644–651.
- [11] B. Wickremasinghe, R. N. Calheiros, and R. Buyya, "CloudAnalyst: A cloudsims-based visual modeller for analysing cloud computing environments and applications," in *Proceedings - International Conference on Advanced Information Networking and Applications, AINA*, 2010, pp. 446–452.
- [12] S. Mostinckx, T. Van Cutsem, S. Timmermont, E. G. Boix, É. Tanter, and W. De Meuter, "EMUSIM: an integrated emulation and simulation environment for modeling, evaluation, and validation of performance of Cloud computing applications," *Softw. - Pract. Exp.*, vol. 39, no. April 2012, pp. 661–699, 2009.
- [13] F. Fittkau, S. Frey, and W. Hasselbring, "CDOSim: Simulating cloud deployment options for software migration support," in *2012 IEEE 6th International Workshop on the Maintenance and Evolution of Service-Oriented and Cloud-Based Systems, MESOCA 2012*, 2012, pp. 37–46.
- [14] J. Jung and H. Kim, "MR-CloudSim: Designing and implementing MapReduce computing model on CloudSim," in *International Conference on ICT Convergence*, 2012, pp. 504–509.
- [15] D. Kliazovich and P. Bouvry, "DENS : Data Center Energy-Efficient Network-Aware Scheduling."
- [16] D. Kliazovich, P. Bouvry, and S. U. Khan, "Simulating communication processes in energy-efficient cloud computing systems," *2012 1st IEEE Int. Conf. Cloud Networking, CLOUDNET 2012 - Proc.*, pp. 215–217, 2012.
- [17] M. Tighe, G. Keller, M. Bauer, and H. Lutfiyya, "DCSim : A Data Centre Simulation Tool for Evaluating Dynamic Virtualized Resource Management," in *Network and service management (cnsm), 2012 8th international conference and 2012 workshop on systems virtualization management (svm)*, 2012, pp. 385–392.
- [18] S. H. Lim, B. Sharma, G. Nam, E. K. Kim, and C. R. Das, "MDCSim: A multi-tier data center simulation platform," in *Proceedings - IEEE International Conference on Cluster Computing, ICC*, 2009.
- [19] "CSIM Development Toolkit for Simulation and Modeling." [Online]. Available: <http://www.mesquite.com/>. [Accessed: 24-Apr-2015].
- [20] S. Sotiriadis, N. Bessis, N. Antonopoulos, and A. Anjum, "SimIC: Designing a new Inter-Cloud simulation platform for integrating large-scale resource management," in *Proceedings - International Conference on Advanced Information Networking and Applications, AINA*, 2013, pp. 90–97.
- [21] S. Ostermann, K. Plankensteiner, R. Prodan, and T. Fahringer, "GroudSim: An event-based simulation framework for computational grids and clouds," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2011, vol. 6586 LNCS, no. 261585, pp. 305–313.
- [22] I. Sriram, "SPECI, a simulation tool exploring cloud-scale data centres," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2009, vol. 5931 LNCS, pp. 381–392.
- [23] W. John and S. Tafvelin, "Analysis of internet backbone traffic and header anomalies observed," *IMC '07 Proc. 7th ACM SIGCOMM Conf. Internet Meas.*, pp. 111–116, 2007.
- [24] M. Al-Fares, S. Radhakrishnan, B. Raghavan, N. Huang, and a Vahdat, "Hedera: Dynamic flow scheduling for data center networks," *Proc. 7th USENIX Conf. Networked Syst. Des. Implement.*, p. 19, 2010.
- [25] S. Kandula, S. Sengupta, A. Greenberg, P. Patel, and R. Chaiken, "The Nature of Datacenter Traffic : Measurements & Analysis," *Provider*, no. Microsoft, pp. 202–208, 2009.
- [26] T. Benson, a Anand, a Akella, and M. Zhang, "The case for fine-grained traffic engineering in data-centers," *Proc. INM/WREN'10*, 2010.

- [27] H. Wang, H. Xie, L. Qiu, and Y. Yang, "COPE: traffic engineering in dynamic networks," *Acm Sigcomm ...*, pp. 99–110, 2006.
- [28] T. Benson, A. Anand, A. Akella, and M. Zhang, "Understanding data center traffic characteristics," *ACM SIGCOMM Comput. ...*, vol. 40, no. 1, pp. 92–99, 2010.
- [29] T. Benson, A. Anand, A. Akella, and M. Zhang, "MicroTE: fine grained traffic engineering for data centers," *Proc. Seventh Conf. Emerg. Netw. Exp. Technol. - Conex. '11*, pp. 1–12, 2011.
- [30] K. Bilal, S. U. R. Malik, S. U. Khan, and A. Y. Zomaya, "Trends and challenges in cloud datacenters," *IEEE Cloud Comput.*, vol. 1, no. 1, pp. 10–20, 2014.
- [31] R. Niranjana Mysore, A. Pamboris, N. Farrington, N. Huang, P. Miri, S. Radhakrishnan, V. Subramanya, A. Vahdat, and R. N. Mysore, "PortLand: a scalable fault-tolerant layer 2 data center network fabric," *SIGCOMM '09 Proc. ACM SIGCOMM 2009 Conf. Data Commun.*, pp. 39–50, 2009.
- [32] D. D. T. C. P. D. Tcp and B. Vamanan, "Deadline-Aware Datacenter TCP (D<sup>2</sup>TCP)," *Sigcomm*, pp. 115–126, 2012.
- [33] C. Wilson, H. Ballani, T. Karagiannis, and A. Rowstron, "Better Never than Late: Meeting Deadlines in Datacenter Networks," *Proc. ACM Conf. Commun. Archit. Protoc. Appl.*, pp. 50–61, 2011.
- [34] M. Alizadeh, A. Greenberg, D. a. Maltz, J. Padhye, P. Patel, B. Prabhakar, S. Sengupta, and M. Sridharan, "Data center TCP (DCTCP)," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 40, no. 4, p. 63, 2010.
- [35] M. Alizadeh, T. Edsall, S. Dharmapurikar, R. Vaidyanathan, K. Chu, A. Fingerhut, V. The, L. Google, F. Matus, R. Pan, N. Yadav, and G. V. Microsoft, "CONGA : Distributed Congestion-Aware Load Balancing for Datacenters," *Proc. ACM SIGCOMM 2014 Conf. SIGCOMM - SIGCOMM '14*, pp. 503–514, 2014.
- [36] H. Chourabi, J. R. Gil-garcia, T. A. Pardo, H. J. Scholl, S. Walker, and K. Nahon, "Understanding Smart Cities : An Integrative Framework," 2012.
- [37] R. E. Hall, J. Braverman, J. Taylor, and H. Todosow, "The Vision of A Smart City," in *2nd International Life Extension Technology Workshop, 09/28/2000--09/28/2000;*, 2000.