Sustainable Availability Provision in Distributed Cloud Services

Olga V. Yanovskaya, Max E. Yanovsky, and Vyacheslav S. Kharchenko
Computer Systems and Networks Department
National Aerospace University "KhAI"
Kharkiv, Ukraine
e-mail: {O.Yanovskaya, M.Yanovsky, V.Kharchenko}@csn.khai.edu

Ah-Lian Kor, and Colin Pattinson
School of Computing, Creative Technologies &
Engineering
Leeds Beckett University
Leeds, United Kingdom
e-mail: {A.Kor, C.Pattinson}@leedsbeckett.ac.uk

***Abstract* — The article is an extension of this paper[1]. It describes methods for dealing with reliability and fault tolerance issues in cloud-based datacenters. These methods mainly focus on the elimination of a single point of failure within any component of the cloud infrastructure, availability of infrastructure and accessibility of cloud services. Methods for providing the availability of hardware, software and network components are also presented. The analysis of the actual accessibility of cloud services and the mapping of a cloud-based datacenter infrastructure with different levels of reliability to the Tier Classification System[2] is described. Non-compliance of the actual accessibility with the level of High Availability for cloud web services is unraveled.**

Keywords- Availability, Accessibility, Cloud Architectures, Service Reliability, Fault Tolerance

I. Introduction

According to IBM, high availability is a critical issue for cloud computing[3]. Before new technology is introduced to the market, a number of factors need to be considered – timing, market maturity, competition, economic affairs etc. Many open source software platforms are based on cloud technologies which facilitate the development and introduction of new technologies to a wider audience.

Presenting new technologies as an additional component of an existing platform provides developers with the opportunity to introduce their technology gradually to the market. This derives from a number of reactions observed in the market, such as caution, suspicion, security concerns or even the lack of openness towards new processes, operations, and new tools. Gradual release of a product or service will enable customers-users have extra time to process the new information, learn more about the new technology, and make necessary functional and technological changes within their organizations so that it will not impede the smooth ing of the organization. A vital property is failure-free operation, a property of an object that refers to permanent operability during some period of time. Time to first failure is a property which is characterized by the probability of failure-free operation, or in other words, likelihood of absence of failure within a given operating time. Moreover, it is important to consider cloud technology within the frame of sustainability relating to environmental impact and energy efficiency. Thus, the goal of the article is to define the ways for sustainable availability provision in cloud systems with client-server and distributed peer-to-peer (P2P) based architectures.

II. State of the Art

Studies' analysis [1-3] leads to the conclusion that currently, there is ambiguous interpretation of cloud-based datacenter operability conditions as it depends on the number of available and unavailable services in relation to the total amount of services, at current point in time. The term of cloud service accessibility is used, viewing the fact that the end user interacts with a specific cloud-based datacenter service. Figure 1 depicts two essential quality criteria of the services: reliability (i.e.

[1] http://www.pacet.gr/Papers/PACET_2015_submission_72.pdf
[2] http://www.gpxglobal.net/wp-content/uploads/2012/10/TIERSTANDARD_Topology_120801.pdf
[3] http://www-935.ibm.com/services/za/gts/cloud/Security_and_high_availability_in_cloud_computing_environments.pdf

failure-free operation and availability); performance (operativeness). Note that accessibility transcends across reliability and performance.

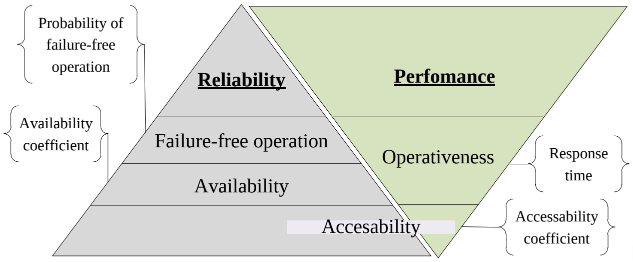

Figure. 1. Correlation between availability, reliability and accessibility indicators

Analysis of the sources [4, 5] shows that the most common availability indicator is determined by the following formula:

$$K_a = MTTF/(MTTF + MTTR), \tag{1}$$

where K_a – availability coefficient,
$MTTF$ – mean time to failure,
$MTTR$ – mean time to recover.

Property of accessibility determines the probability that at any point in time, a certain cloud service will be available to the end user with a satisfactory response time. The main factor is the accessibility coefficient, which includes not only availability, but also the functional properties of the system.

With increasing demands on the quality of services of the IT-infrastructure, any kind of failure in the network is unacceptable. Even a relatively small packet loss can have a negative impact on the end-users' quality of service, especially for critical and business-critical processes. This means the failure of the main switching node, link or interface may have serious consequences for the provider. The design of the cloud-based datacenter should help minimize network failures and the severity of the consequences of potential accidents. Advances in technology and the pace of the construction of virtual data centers as well as cloud infrastructures have brought about the following: development of requirements for the distribution functions of management control across multiple geographically dispersed nodes; division of responsibility between distributed teams of technical personnel; extension of monitoring and diagnostics functions support high availability and disaster recovery. According to [6], cloud-based datacenter design should include redundant components and distributed platforms, so that the physical connection and access to resources remain constant, regardless of the location and value of the current availability and performance indicators. Furthermore, to protect the competitiveness of enterprises and organizations that are customers of cloud providers, critical business applications need to be available 24/7. In case of environmental or technological disasters, the data must be restored with minimal disruption, calling for an emergency backup and recovery of business applications and the virtual machine in a different availability zone will ensure that user data is protected and accessible from anywhere. Typically, network architects predict a 4 or 5 "nines" system availability [6]. However, each additional digit = "9" can significantly increase the cost of deployment. To achieve near-zero downtime per year of the cloud-based datacenter, one must consider not only the reliability of the hardware and network infrastructure, but also part of software.

III. Methods for providing availability of hardware components within the client-server cloud architecture

The objective of this collection of methods is to maintain the availability and sustainability of cloud services as well as applications in the event a particular server becomes unavailable. These methods can operate at multiple levels within the datacenter infrastructure[4]. Hardware component accessibility methods are used at the physical layer ISO/OSI model. These include the following.

• *Grouping of network adapters and communication channels.* In order to eliminate single points of failure at the level of communication within a network, access layer servers have multiple (two or more) network interfaces. This method is named NIC-Teaming[5] and it involves the grouping of multiple physical connections into one logical channel - LAG (Link Aggregation). The logical connection may be in active-active mode that combines multiple channels into a single logical load sharing or active-passive mode, while the second interface is idle as long as the first interface operates as usual.

[4] http://www.gpxglobal.net/wp-content/uploads/2012/10/TIERSTANDARD_Topology_120801.pdf
[5] https://technet.microsoft.com/en-us/library/hh831648(v=ws.11).aspx

• *Using hot-swappable interfaces*[6]. This method entails installation or removal of the interface card on the router or switch without having to power off the device. The controller dynamically recognizes the new interface and begins the data exchange. As a result, new components can be inserted and removed without interrupting the system's operation.

• *Use of highly reliable server access layer.* Hardware components of highly reliable servers have the highest values of MTTF.

IV. PROVIDING AVAILABILITY OF SOFTWARE COMPONENTS WITHIN THE CLIENT-SERVER CLOUD ARCHITECTURE

An analysis [10, 11] shows that the main methods of sustainable resilience at the application level are: the use of pooling resources; protected applications and resources of critical applications; the migration of virtual machines and the use of Unified In-Service Software Upgrades. A detailed discussion of these methods are as follows:

• *Using application pools of resources.* This method is based on the fact that multiple instances of applications are combined to the pool of resources that are distributed throughout the network (Fig. 2). According to [11], the use of resource pools is an effective solution for resiliency, but the main disadvantage of this approach is the problem of synchronization. This solution requires more effective methods of planning, synchronization and load balancing on the coordination sites.

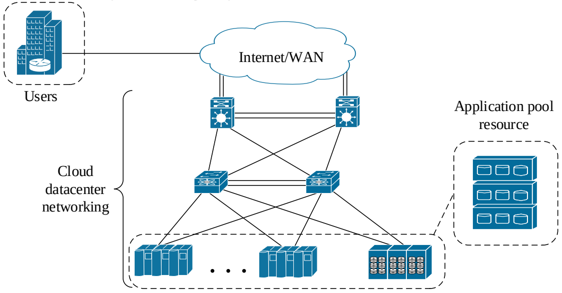

Figure. 2. Use of application pools of resources

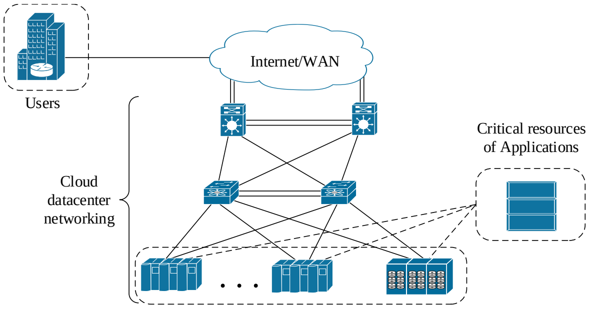
Internet/WAN
Users
Critical resources of Applications
Cloud datacenter networking
. . .

Figure. 3. Transfer of critical resources of applications

• *Transfer of critical application resources.* Some applications have critical resources which for various reasons, are not possible or desirable to replicate. They can either work on high-performance servers that makes replication too expensive, or they can include critical resources that makes replication impossible due to security threats or exploitation reasons. Under these conditions, one application server is a single point of failure. In order to minimize the risk of failure for critical resources of applications, it will require the following: applications are executed in several powerful servers; failover and high availability provided by the active/standby configuration mode for disaster recovery. Connection problems are solved by multi-session network connections among the server and clients, as well as multiple network routes (Fig. 3). Conditions necessary for an effective deployment of this method are: the presence of redundant network links and backup systems; continuous monitoring of the status of servers; optimal data replication; synchronization of the active and standby systems.

• *Migration of virtual machines*. Virtual machine migration is an effective method for: facilitating fault-tolerance; improving energy efficiency; and maintaining service availability in the event of a failure of the physical server which hosts the VMs. This method assumes that the virtual machine has its own running copy on a server located in another rack or in another datacenter. In this case, services that are deployed on the initial virtual machine are replicated on another virtual machine.

• *Using a single integrated service system updates.* The ability to provide a unified system ISSU (Unified In-Service Software Upgrades) updates of an operating system without shutting down network devices (that are scheduled for preliminary verification of compatibility) is supported by some versions of operating systems within a number of network

[6] https://www.ibm.com/support/knowledgecenter/SSPHQG_6.1.0/com.ibm.hacmp.admngd/ha_admin_replace_pcihotplug_nic.htm

equipment manufacturers [11]. This will help avoid risks associated with downtime and failed updates of network operating systems.

V. PROVIDING AVAILABILITY OF NETWORK COMPONENTS WITHIN THE CLIENT-SERVER CLOUD ARCHITECTURE

• *Redundant network devices.* The analysis of the examined standards and guidelines for the design of a datacenter reveals that network devices redundancy as a method to foster fault tolerance, involves the duplication of the core level routers, access layer and distribution switches. Additional mechanisms for balancing the load between them increase network performance and reduce latency. Apart from that, in order to minimize the effects of a single point of failure, the network device may also be used in methods such as hot-swappable interface, Unified In-Service Software Upgrades, redundant switching and routing mechanisms.

• *Redundant switching and routing mechanisms*. The main purpose of this method is to create redundant switching for network devices. In addition to a redundant configuration, switching fabric with two switch modules is used to increase the capacity and performance of the switch. The third module, if present, provides an additional precision (2 + 1) for switching functions, so that if one of the two functional modules becomes inoperable, a third module can assume the function of the failed module. Redundant routing mechanisms provide simultaneous operation of multiple routing protocols as well as protection from the routing and switching loops. A list of relevant protocols, technology and standards is as follows:

- L3 dynamic routing protocols in the core level (OSPF, RIP, or static routing);
- Multiple Spanning Tree Protocol (MSTP);
- MPLS in the core level;
- 802.3ad LAG;
- 802.1q Virtual LANs;
- RTG (Redundant trunk groups);
- VRRP;
- MPLS in the aggregation level.

The previously discussed methods have a number of common disadvantages in their implementation. They are: complexity of the architecture; the processes of its maintenance and operation due to demand for high priced resources; additional overhead costs on excess equipment and permanent high-quality maintenance. In order to determine the effectiveness of the methods and architectures considered, an analysis of the accessibility of services from known cloud providers should be conducted.

VI. ANALYSIS OF ACTUAL SERVICES ACCESSIBILITY OF CLOUD PROVIDERS

Studies' analysis [2, 3] leads to the view that currently, there is ambiguous interpretation of cloud-based datacenter operability conditions. Figures 4 and 5 are bar graphs that illustrate the results of statistical data analyses conducted on services accessibility of various cloud providers.

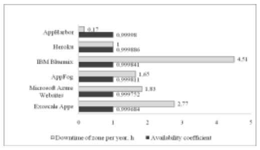

Figure. 4. Actual services accessibility of cloud providers: PaaS service model

The values shown in the histograms (Fig. 4 and 5) are obtained by analyzing the actual values of downtime for a time period of 1 year, as published by Cloud Harmony [12] for PaaS and IaaS service models [13]. Based on the analysis of the actual accessibility of cloud providers' services, we can conclude that the average accessibility of a datacenter's cloud services corresponds to a value of 0.999.

In order to determine whether the claimed quality of the service is in compliance with the actual quality of service, an analysis of the service-level agreement (SLA) of known cloud providers is performed. The results of the analysis are summarized in the Table 1.

In order to verify compliance of the datacenter infrastructure cloud providers to levels of reliability (in the Tier Classification System), a data analysis has been conducted and results of the analysis are presented in the Table 2.

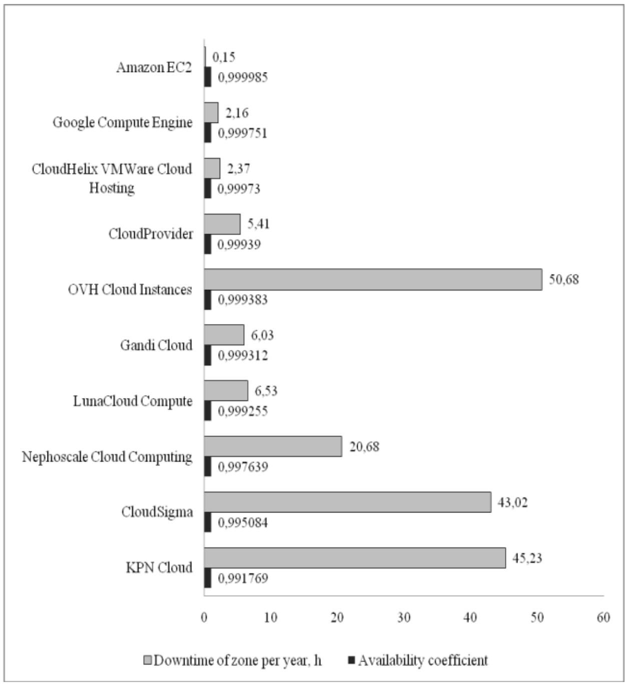

Figure. 5. Actual services accessibility of cloud providers: IaaS service model

TABLE 1. RESULTS OF THE COMPARATIVE ANALYSIS OF THE DATACENTER INFRASTRUCTURE RELIABILITY LEVELS

Cloud provider / Service type	**Claimed levels of service accessibility (replication services provided in least two availability zones)**
Microsoft Azure / Microsoft Online Services [14]	99.95%
Microsoft Azure / Virtual Machines [15]	99.95%
Microsoft Azure / Cloud Services [16]	99.95%
Google Cloud Platform / Google Compute Engine [17]	99.95%
Google Cloud Platform / Google App Engine [18]	99.95%
Amazon EC2 [19]	99.95%
Rackspace Cloud Servers [20]	99.90%

TABLE 2. RELIABILITY LEVELS OF CLOUD PROVIDERS' DATACENTERS

Cloud provider	**Tier Reliability Level**
Amazon [21]	IV
Microsoft Azure [22]	IV
Rackspace Cloud [23]	IV

Analysis of the results leads to the following conclusion: despite the fact that the cloud datacenter infrastructure matches the fourth level of reliability with an availability coefficient of 0.99995, the actual average access-infrastructure of cloud service providers, on the average, corresponds to a value of 0.999. Therefore, it is necessary to improve the models and methods of assessing the availability and accessibility of services provided through a client-server cloud infrastructure in order to obtain more accurate estimates of the reliability indices.

VII. PROVIDING AVAILABILITY WITHIN A DISTRIBUTED CLOUD ARCHITECTURE

A list of methods that addresses availability of the cloud is as follows:

• *Passive replication*. Passive replication ensures service availability in a distributed Cloud network. The availability level of the service is determined by the number of nodes that could provide the resource or part of the resource requested by users. The features of passive replication include the entire replication process as a self-organized procedure, which occurs automatically without a user's intervention or a centralized management system administrator. Consequently, this will help eliminate any point of failure.

- *Using the distributed structure of DNS servers*. Name servers are an essential part of cooperation between nodes in the distributed Cloud. The process of resource publication and replication occurs through the changes of the DNS server's resource records. DNS servers worldwide are joined together to form a single distributed overlay network.
- *Application of the Distributed Hash Table (DHT)*. The nodes operate based on the Kademlia protocol[7] and communicate with each other using the transport layer of the OSI model protocol UDP (User Datagram Protocol). Its nodes store data using DHT. The algorithm is based on the calculated "distance" between the nodes by applying the XOR operation to the ID of the node. DHT includes mapping of the hash function, matching identifiers (names) with resources and keys. Eliminating a centralized tracker server allows the network to be fault-tolerant, even in cases with dramatic changes in the participants' number.
- *Use of always-on backup server*. A different method that is proposed for providing a fault tolerance architecture for a decentralized Cloud is the implementation of an infrastructure with backup always-on server that will be in a standby mode 24/7. This ensures an uninterrupted operation of the service in cases where all nodes are turned off or the replication resource is not sufficiently distributed among the nodes. In situations where nodes have the full resource or part of it are missing, the node that first requests a service, can receive a response from the always-on server. Applying always-on backup server not only provides a system redundancy property but also allows load balancing among the nodes in a distributed Cloud.

VIII. Availability model of distributed cloud service

In order to create a model, methods of analytical modeling algorithms for behavior of discrete-continuous stochastic systems[8] are used and they are based on the basic events and structural-automaton models (SAM) [25]. A list of basic events used for this research is tabulated in Table 3.

Figure 6 shows an availability model of distributed cloud service with redundant always-on server. It is drawn based on values tabulated in Table 4.

Table 3. Basic events

Basic Events	Description
BE1	achieve a satisfactory level of the resource's part replication
BE2	failure of the resource node owner
BE3	failure of the backup server
BE4	decrease the resource replication level
BE5	finalize the resource owner node recovery procedures
BE6	finalize the backup server recovery process

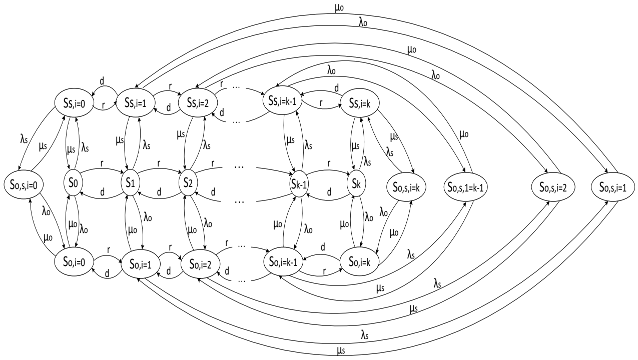

Figure 6. Availability Model of distributed cloud Service with redundant always-on server

The states of the model in Figure 6 and Table 4 are explained below:

- S_0 – the initial state of the system in which the only available user node is the resource owner and the backup server;
- S_j - where j =1, … , k – states in which the owner of the resource, the backup server and i parts of the resources are available to the user node with a satisfactory level of replication;
- $S_{o,i=j}$, - where j =1, … , k – states of the failure mode of the resource owner node, where the backup server and i part of the resources are available with a satisfactory level of replication;
- $S_{s,i=j}$, where j =1, … , k – state of backup server failure, where the resource owner and i parts of the resource are available to the user node with a satisfactory level of replication;
- $S_{o,s,i=j}$, where j =1, … , k – state of backup server and owner of the resource failure, where i pieces of the resource are available with a satisfactory level of replication.

[7] https://pdos.csail.mit.edu/~petar/papers/maymounkov-kademlia-lncs.pdf
[8] http://people.eecs.berkeley.edu/~ananth/223Fall08/Textbook.pdf

Parameters of the model are as follows:

- k – number of similar (identical in size and intensity to the user requests) part of the resource;
- λ_o – the failure rate of the resource owner;
- μ_o – the recovery rate of the resource owner;
- λ_s – the failure rate of the backup server;
- μ_s – the recovery rate of the backup server;
- r – the replication rate of the resource part with a satisfactory degree (the inverse of the average time replication of the resource to a satisfactory degree);
- d – reduction rate of the resource replication to being unsatisfactory (the inverse of the average time of reducing the resource replication to unsatisfactory).

TABLE 4. STATE MODEL

State and current base event	Current value of the state vector			Condition	Transition	Rate
	V_o	V_s	V_i			
–	1	1	0	S_0	–	–
S_0BE1	1	1	1	S_1	$S_0 \rightarrow S_1$	R
S_0BE2	0	1	0	$S_{o,i=0}$	$S_0 \rightarrow S_{0,i=0}$	Λo
S_0BE3	1	0	0	$S_{s,i=0}$	$S_0 \rightarrow S_{s,i=0}$	Λs
S_1BE1	1	1	2	S_2	$S_1 \rightarrow S_2$	R
S_1BE2	0	1	1	$S_{o,i=1}$	$S_1 \rightarrow S_{0,i=1}$	Λo
S_1BE3	1	0	1	$S_{s,i=1}$	$S_1 \rightarrow S_{s,i=1}$	Λs
S_1BE4	1	1	0	S_0	$S_2 \rightarrow S_0$	D
S_2BE1	1	1	3	S_3	$S_2 \rightarrow S_2$	R
S_2BE2	0	1	2	$S_{o,i=2}$	$S_2 \rightarrow S_{0,i=2}$	λ_o
S_2BE3	1	0	2	$S_{s,i=2}$	$S_2 \rightarrow S_{s,i=2}$	λ_s
S_2BE4	1	1	1	S_1	$S_2 \rightarrow S_1$	D
S_{k-1}BE1	1	1	k	S_k	$S_{k-1} \rightarrow S_k$	r
S_{k-1}BE2	0	1	k-1	$S_{o,i=k-1}$	$S_{k-1} \rightarrow S_{0,i=k-1}$	λ_o
S_{k-1}BE3	1	0	k-1	$S_{s,i=k-1}$	$S_{k-1} \rightarrow S_{s,i=k-1}$	λ_s
S_{k-1}BE4	1	1	k-2	S_{k-2}	$S_{k-1} \rightarrow S_{k-2}$	d
S_kBE2	0	1	k	$S_{o,i=k}$	$S_{k-1} \rightarrow S_{0,i=k-1}$	λ_o
S_kBE3	1	0	k	$S_{s,i=k}$	$S_{k-1} \rightarrow S_{s,i=k-1}$	λ_s
S_kBE4	1	1	k-1	S_{k-1}	$S_{k-1} \rightarrow S_{k-2}$	d
$S_{0,i=0}$BE1	0	1	1	$S_{o,i=1}$	$S_{0,i=0} \rightarrow S_{0,i=1}$	r
$S_{0,i=0}$BE3	0	0	0	$S_{o,s,i=0}$	$S_{0,i=0} \rightarrow S_{0,s,i=0}$	λ_s
$S_{0,i=0}$BE5	1	1	0	S_0	$S_{0,i=0} \rightarrow S_0$	μ_o
$S_{0,i=1}$BE1	0	1	2	$S_{o,i=2}$	$S_{0,i=1} \rightarrow S_{0,i=2}$	r
$S_{0,i=1}$BE3	0	0	1	$S_{o,s,i=1}$	$S_{0,i=1} \rightarrow S_{0,s,i=1}$	λ_s
$S_{0,i=1}$BE4	0	1	0	$S_{o,i=0}$	$S_{0,i=1} \rightarrow S_{0,i=0}$	d
$S_{0,i=1}$BE5	1	1	1	S_1	$S_{0,i=1} \rightarrow S_1$	μ_o
$S_{0,i=2}$BE1	0	1	3	$S_{o,i=3}$	$S_{0,i=2} \rightarrow S_{0,i=3}$	r
$S_{0,i=2}$BE3	0	0	2	$S_{o,s,i=2}$	$S_{0,i=2} \rightarrow S_{0,s,i=2}$	λ_s
$S_{0,i=2}$BE4	0	1	1	$S_{o,i=1}$	$S_{0,i=2} \rightarrow S_{0,i=1}$	d
$S_{0,i=2}$BE5	1	1	2	S_2	$S_{0,i=2} \rightarrow S_2$	μ_o
$S_{o,i=k-1}$BE1	0	1	k	$S_{o,i=k}$	$S_{0,i=k-1} \rightarrow S_{0,i=k}$	r
$S_{o,i=k-1}$BE3	0	0	k-1	$S_{o,s,i=k-1}$	$S_{o,i=k-1} \rightarrow S_{0,s,i=k-1}$	λ_s
$S_{o,i=k-1}$BE4	0	1	k-2	$S_{o,i=k-2}$	$S_{o,i=k-1} \rightarrow S_{0,i=k-2}$	d
$S_{o,i=k-1}$BE5	1	1	k-1	S_{k-1}	$S_{o,i=k-1} \rightarrow S_{k-1}$	μ_o
$S_{o,i=k}$BE3	0	0	k	$S_{o,s,i=k}$	$S_{o,i=k} \rightarrow S_{0,s,i=k}$	λ_s
$S_{o,i=k}$BE4	0	1	k-1	$S_{o,i=k-1}$	$S_{o,i=k} \rightarrow S_{0,i=k-1}$	d
$S_{o,i=k}$BE5	1	1	k	S_k	$S_{o,i=k} \rightarrow S_k$	μ_o
$S_{s,i=0}$BE1	1	0	1	$S_{o,i=1}$	$S_{s,i=0} \rightarrow S_{o,i=1}$	r
$S_{s,i=0}$BE2	0	0	0	$S_{o,s,i=0}$	$S_{s,i=0} \rightarrow S_{o,s,i=0}$	λ_o
$S_{s,i=0}$BE6	1	1	0	S_0	$S_{s,i=0} \rightarrow S_0$	μ_s
$S_{s,i=1}$BE1	1	0	2	$S_{s,i=2}$	$S_{s,i=1} \rightarrow S_{s,i=2}$	r
$S_{s,i=1}$BE2	0	0	1	$S_{o,s,i=1}$	$S_{s,i=1} \rightarrow S_{0,s,i=1}$	λ_o
$S_{s,i=1}$BE4	1	0	0	$S_{s,i=0}$	$S_{s,i=1} \rightarrow S_{s,i=0}$	d
$S_{s,i=1}$BE6	1	1	1	S_1	$S_{s,i=1} \rightarrow S_1$	μ_s
$S_{s,i=2}$BE1	1	0	3	$S_{s,i=3}$	$S_{s,i=2} \rightarrow S_{s,i=3}$	r
$S_{s,i=2}$BE2	0	0	2	$S_{o,s,i=2}$	$S_{s,i=2} \rightarrow S_{0,s,i=2}$	λ_o
$S_{s,i=2}$BE4	0	1	1	$S_{s,i=1}$	$S_{s,i=2} \rightarrow S_{0,i=1}$	d
$S_{s,i=2}$BE6	1	1	2	S_2	$S_{0,i=2} \rightarrow S_2$	μ_s

$S_{s,i=k-1}$BE1	1	0	k	$S_{s,i=k}$	$S_{s,i=k-1} \rightarrow S_{s,i=k}$	r
$S_{s,i=k-1}$BE2	0	0	k-1	$S_{o,s,i=k-1}$	$S_{s,i=k-1} \rightarrow S_{o,s,i=k-1}$	λ_o
$S_{s,i=k-1}$BE4	1	0	k-2	$S_{s,i=k-2}$	$S_{s,i=k-1} \rightarrow S_{s,i=k-2}$	d
$S_{s,i=k-1}$BE6	1	1	k-1	S_{k-1}	$S_{s,i=k-1} \rightarrow S_{k-1}$	μ_o
$S_{s,i=k}$BE2	0	0	k	$S_{o,s,i=k}$	$S_{s,i=k} \rightarrow S_{0,s,i=k}$	λ_o
$S_{s,i=k}$BE4	1	0	k-1	$S_{s,i=k-1}$	$S_{s,i=k} \rightarrow S_{0,i=k-1}$	d
$S_{s,i=k}$BE6	1	1	k	S_k	$S_{s,i=k} \rightarrow S_k$	μ_o
$S_{o,s,i=0}$BE5	1	0	0	$S_{s,i=0}$	$S_{o,s,i=0} \rightarrow S_{s,i=0}$	μ_o
$S_{o,s,i=0}$BE6	0	1	0	$S_{o,i=0}$	$S_{o,s,i=0} \rightarrow S_{o,i=0}$	μ_s
$S_{o,s,i=1}$BE5	1	0	1	$S_{s,i=1}$	$S_{o,s,i=1} \rightarrow S_{s,i=1}$	μ_o
$S_{o,s,i=1}$BE6	0	1	1	$S_{o,i=1}$	$S_{o,s,i=1} \rightarrow S_{o,i=1}$	μ_s
$S_{o,s,i=2}$BE5	1	0	2	$S_{s,i=2}$	$S_{o,s,i=2} \rightarrow S_{s,i=2}$	μ_o
$S_{o,s,i=2}$BE6	0	1	2	$S_{o,i=2}$	$S_{o,s,i=2} \rightarrow S_{o,i=2}$	μ_s
$S_{o,s,i=k-1}$BE5	1	0	k-1	$S_{s,i=k-1}$	$S_{o,s,i=k-1} \rightarrow S_{s,i=k-1}$	μ_o
$S_{o,s,i=k-1}$BE6	0	1	k-1	$S_{o,i=k-1}$	$S_{o,s,i=k-1} \rightarrow S_{o,i=k-1}$	μ_s
$S_{o,s,i=k}$BE5	1	0	k	$S_{s,i=k}$	$S_{o,s,i=k} \rightarrow S_{s,i=k}$	μ_o
$S_{o,s,i=k}$BE6	0	1	k	$S_{o,i=k}$	$S_{o,s,i=k} \rightarrow S_{o,i=k}$	µs

TABLE 5. STRUCTURAL AUTOMATON MODEL

Basic event (BE)	**Terms and conditions**	**Formula to calculate intensity of BE**	**Modification rules for state vector components**
BE1	$V_i < k$	r	$V_i :=$ Vi+1
BE2	$V_o = 1$	λ_o	$V_o :=0$
BE3	$V_s = 1$	λ_s	$V_s := 0$
BE4	$V_i > 0$	d	$V_i := V_i – 1$
BE5	$V_o = 0$	μ_o	$V_o :=1$
BE6	$V_s = 0$	μ_s	$V_s := 1$

For the analysis of the proposed model, it is recommended to use the mathematical apparatus of Semi-Markov processes [33]. Table 5 provides details for the structural-automaton model. The details relate to terms and conditions, formula to calculate the intensity for each basic event and modification rules for the state vector components.

IX. CONCLUSION

With technological and scientific innovation, more people and enterprises have access to the Internet, adding to the constantly increasing demand on cloud computing. Their activities generate data through a number of devices, at an accelerating rate and at highly unpredictable moment in time. It is necessary for this generated data to be stored and handled securedly. Additionally, retrieval has to be highly responsive (with low latency) to user requests. To accommodate this requirement large datacenters with super high performance servers have been built. These are big, expensive, energy hungry facilities that only powerful large corporates can afford to have. Undeniably, they require high maintenance costs, personnel resources, powerful backup, and storage space. Datacenters are located at a physical place so they are vulnerable to local conditions, be it weather phenomena, regional power cuts, earthquakes etc... Furthermore, 2% of the global CO_2 emissions is attributed to the ICT industry, a significant part of which is caused by energy consumed by the functioning of the datacenter physical, mechanical, and computing systems. The high cost of investment necessary for datacenters prevents smaller companies from entering the market. Basic methods for infrastructure availability and service accessibility in a sustainability framework for the client-server distributed cloud architectures have been described in this paper.

P2P technology can have an immense benefit on the environment as it significantly reduces CO_2 emissions that come from datacenters' high power demand [34]. By integrating datacenters infrastructures with P2P cloud technology will eventually bring about a new era of sophisticated IT infrastructures and systems with minimal environmental impact.

REFERENCES

[1]. I. Elyasi-Komari, A. Gorbenko, V. Kharchenko, A. Mamalis, “Analysis of Computer Network Reliability and Criticality: Technique and Features”. IJCNS, 4(11), 2011, pp. 720--726. doi: 10.4236/ijcns.2011.411088

[2]. V. Kharchenko, V. Sklyar, and A. Siora, “Dependability of Safety-Critical Computer Systems through Component-Based Evolution”, Proceedings of the 4th International Conference on Dependability of Computer Systems (DepCoS), Brunow, Poland, 2009, pp. 42-49

[3]. K. Dong Seong, F. Machida, and K. S. Trivedi, “Availability Modeling and Analysis of a Virtualized System”, Proceedings of the 15th IEEE Pacific Rim International Symposium on Dependable Computing (PRDC), Shanghai, China, 2009, pp. 365-371.

[4]. S. Fernandes, E. Tavares, M. Santos, V. Lira, P. Maciel, “Dependability assessment of virtualized networks”, IEEE International Conference on Communications (ICC), 2012, pp. 2711--2716

[5]. A. Undheim, A. Chilwan, and P. Heegaard, “Differentiated Availability in Cloud Computing SLAs”, Proceedings of the 12th IEEE/ACM International Conference on Grid Computing (GRID), 2011, pp 129-136.

[6]. TIA/EIA-942: Telecommunications Infrastructure Standard for Data Centers, 2005

[7]. Data Center Site Infrastructure Tier Standart: Topology,

[8]. https://uptimeinstitute.com/publications/asset/tier-standard-topology

[9]. Data Center Site Infrastructure Tier Standart: Topology,

[10]. https://uptimeinstitute.com/publications/asset/tier-standard-operational-sustainability

[11]. About Uptime Institute, https://uptimeinstitute.com/about-ui

[12]. Cisco Data Center Infrastructure 2.5 Design Guide,

[13]. http://www.scn.rain.com/~neighorn/PDF/Cisco_Data_Center_Infrastructure_Design_Guide.pdf

[14]. Juniper Networks, "Cloud Ready Data Center Network Design Guide",

[15]. http://www.juniper.net/us/en/local/pdf/design-guides/8020014-en.pdf

[16]. Research and compare cloud providers and services, https://cloudharmony.com/status

[17]. Cloud Computing Vendors Taxonomy, http://cloudtaxonomy.opencrowd.com/taxonomy

[18]. SLA for App Service, https://azure.microsoft.com/en-us/support/legal/sla/app-service

[19]. SLA for Virtual Machines, https://azure.microsoft.com/en-us/support/legal/sla/virtual-machines

[20]. SLA for Cloud Services, https://azure.microsoft.com/en-us/support/legal/sla/cloud-services

[21]. Google Compute Engine Service Level Agreement, https://cloud.google.com/compute/sla

[22]. Google App Engine Service Level Agreement, https://cloud.google.com/appengine/sla

[23]. Amazon EC2 Service Level Agreement, https://aws.amazon.com/ru/ec2/sla

[24]. Cloud Service Level Agreement, https://www.rackspace.com/information/legal/cloud/sla

[25]. Amazon called out over cloud security,

[26]. http://www.techworld.com.au/article/326287/amazon_called_over_cloud_security_secrecy

[27]. Microsoft Cloud Services,

[28]. https://assets.digitalmarketplace.service.gov.uk/documents/93064/4504230568132608-pricing-document.pdf

[29]. About Rackspace. Global Infrastructure and Uptime Guarantee,

[30]. http://www.rackspace.com/about/datacenters

[31]. A. Gorbenko, A Romanovsky, “Time-Outing Internet Services. Security & Privacy”, IEEE, 11(2), 2013, pp. 68--71. doi: 10.1109/MSP.2013.43

[32]. B. Volochiy, O. Mulyak, and V. Kharchenko. “Automated Development of Markovian Chains for Fault-Tolerant Computer-Based Systems with Version-Structure Redundancy”, pp. 1--14, ICTERI. 2015.

[33]. F. Grabski. “Semi-Markov Processes: Applications in System Reliability and Maintenance”, 2015, Elsevier Inc. ISBN: 978-0-12-800518-7.

[34]. L. Sharifi, et. al. “Energy Efficiency Dilemma: P2P-cloud vs.

[35]. Datacenter”, CloudCom2014, http://www.gsd.inesc-id.pt/~lveiga/papers/LSharifi-Veiga-CloudCom2014.pdf