Optimal Redundancy Design Of Stochastic LAN Devices Failures In Private Clouds

V.Janardhan Babu,B.Suresh Babu,P.Jayanthi

Abstract— *The cloud computing infrastructure need to be selected with optimal number of devices whose reliabilities are stochastic in nature so that the offered cloud service are uninterrupted. Public and private cloud deployment models differ. Public clouds, such as those from Amazon Web Services or Google Compute Engine, share a computing infrastructure across different users, business units or businesses. Whereas the private cloud infrastructure is to be carefully designed keeping in mind of the budget spent, mean lifetime of the offered service as well as the nature of the devices used to built the cloud. A novel attempt is made to build cloud infrastructure by making use of sdp(stochastic dynamic programming) techniques to offer reliable services.*

Index Terms — Compute Engine, Cloud Computing, Private Cloud, Public Cloud, Reliability, SDP, Monte Carlo Simulation.

1 Introduction

THE basic requirements of cloud are 1)A computer 2) Static IP (if you want to access outside the LAN) 3) Linux OS 4) Hard disk (According to your need). Reliability evaluation of such a simple cloud have to be focussed on MTTF / MTTR and CTMC models.

The software testing community has provided precise definition for software reliability. IEEE defines it as "the probability that software will not cause the failure of a system for a specified time under specified conditions"[1]. According to this definition, we define the cloud service reliability as "the probability that cloud service will successfully complete the customer's request for a specified time."

2 Source Of Failure

Considering potential reasons for failure of a service request, we identify two main sources of failure, the first one is software faults on the one hand, and the second one is physical-resource breakdowns. A cloud computing system can schedule a request and divide it into different subtasks and assign the subtasks to different computing resources that may access different data resources over the Internet. The subtasks are actually software programs running on different computing resources, which contain software faults. Physical-resource breakdowns mainly happen due to hardware wear out and represent inevitability in cloud computing systems. Typically, a broken-down resource, which might be a CPU, memory, or a network link, is eventually repaired or replaced by a functionally equivalent new resource [3]. The reliability of such resources is expressed in terms of Mean Time to Failure (MTTF) and Mean Time to Repair (MTTR). We assume that any service request that uses a certain resource r at some point in time t fails if resource is broken-down at t. Due to these Sources of Failure, it is obvious that a new hybrid model has to be developed for cloud reliability. Based on the above proposal the system reliability is given by[2]

3 Cloud Service Modelling

In the process of a cloud service, facing all kinds of possible events such as failure, repair and recovery, it gives rise to a wide range of system states which are relevant to its ability to accept and successfully carry out its service. As shown in Figure 1, the states are hierarchically categorized as Running or Failure. A Running state is further classified in to Recovering (RI), Accessible (A) and Inaccessible (IA). When the system is in Recovering state, it can process service requests but not be normal interaction. Inaccessible state is belonging to the Running state. For example, a system can be ready to accept service requests, but the cloud service received a denial of service or external attack, which may cause the fact that the user's service requests cannot arrive at the system.

When the cloud computing system is running, it may be failed caused by various kinds of reasons. For some failures cloud service can tolerate, such as failure within the ability of cloud services tolerance, we consider that cloud service is still in the Running state. For the failure cannot be tolerated, we think cloud services into a Failure state. Here, the failure is divided into the recoverable failure (Ra) and unrecoverable failure (N). In recoverable failure state, the cloud service enters the recovery state through repairing or replacing it. If in unrecoverable state, only restart it can be possible to repair the cloud service.

From the reliability perspective, some of the sojourn time distribution functions may be non-exponential [4]. Therefore, the stochastic model based on the state graph is formu-

lated in terms of a semi-Markov process (SMP), which is a more general model than continuous-time Markov chain (CTMC) commonly used. It is obvious that this SMP model is also feasible when all the distribution functions are exponential, for a Markov process is a special case of SMP. To evaluate the cloud service reliability, we need to analyse the SMP model of the service describe by its state transition diagram. Here, the reliability indicates that the probability is the service doesn't enter Ra and/or N state. So we class the SMP states into absorbing and transient categories depending on the actual nature of the attributes being analyzed. Once the system reaches any of the absorbing states, the probability of moving out of such state is 0. Hence, from a normal initial state, the probability is calculated with the visit count and sojourn time

$$V_A = \frac{1}{1 - P_A P_{IA}}; V_{IA} = \frac{P_A}{1 - P_A P_{IA}}; V_R = 0 \tag{6}$$

We consider that service in any state of {*A*, *IA*, *R*} is in the *Running* state. So we regard the three states as a macro state, then $Q = [1]$. Therefore we have the expression of the sojourn time of the *Running* state, as

$$t_{RE} = \sum_{i \in S_i} V_i t_i = V_A t_A + V_{IA} t_{IA} + V_R t_R = V_A t_A + V_{IA} t_{IA} \tag{7}$$

Hence, the probability of a cloud service in time interval [0, t] is

$$R(t) = \frac{e^{-t/t_{RE}}}{t_{RE}} = \frac{e^{-t/(V_A t_A + V_{IA} t_{IA})}}{V_A t_A + V_{IA} t_{IA}}$$

Assuming that there are K kinds of cloud services in the system and each kind of them have n services. Then for each kind of the service, its reliability is
of each transient states [4].

The SMP corresponding to Figure 2 can be described in terms of its embedded discrete-time Markov chain (DTMC). According to the model and parameters shown above, the DTMC transition probability matrix could be written as:

$$P = \begin{array}{c} \\ A \\ IA \\ R \\ Ra \\ N \end{array} \begin{array}{c} \begin{array}{ccccc} A & IA & R & Ra & N \end{array} \\ \begin{bmatrix} 0 & P_{IA} & 0 & P_{Ra} & P_N \\ P_A & 0 & 0 & P_{Ra} & P_N \\ P_{RA} & P_{AIA} & 0 & 0 & 0 \\ 0 & 0 & P_R & 0 & 0 \\ P_{NA} & P_{NIA} & 0 & 0 & 0 \end{bmatrix} \end{array} \tag{1}$$

Since the probability of transitioning from one state to another state must be 1, we have

$$P_{IA} + P_{Ra} + P_N = 1; P_A + P_{Ra} + P_N = 1;$$
$$P_{RA} + P_{RIA} = 1; P_R = 1; P_{NA} + P_{NIA} = 1$$

The visit count of a cloud service in every state can be computed as $V = VgP$, where $V = [V_A, V_{IA}, V_R, V_{Ra}, V_N]$, in addition with

$$\sum_i V_i = 1, i \in \{A, IA, R, Ra, N\} \tag{2}$$

As mentioned above, we define the cloud service reliability as "the probability that cloud service will successfully complete the customer's request for a specified time." In another words, the reliability could also be defined as the probability that the cloud service will continuously and correctly complete the customer's request in the time interval [0, t]. We assume that the service is in the normal state at time 0 [8,9]. Because of the variety of the distribution of the sojourn time in each state, it is difficult to give a general and normal expression of the probability. But, when the sojourn time conforms to exponential distribution, the sojourn time in the Running state is also exponential, then we have the probability density function f(x) ¼ le 2lx and probability function F(x) ¼ 1 2 le 2lx, l ¼ 1/tRE_tRE is the sojourn time of a service in the Running

$$R(t) = 1 - F(t) = \lambda e^{-\lambda t} = \frac{e^{-t/t_{RE}}}{t_{RE}} \tag{3}$$

Next, we calculate the sojourn time of a service in the Running state. We obtain the fact that is the set of transient states by analyzing the model with the previous description. Where, the transition probabilities between transient states are expressed as

$$Q = \begin{bmatrix} 0 & P_{IA} & 0 \\ P_A & 0 & 0 \\ P_{RA} & P_{RIA} & 0 \end{bmatrix} \tag{4}$$

We assume the initial state here is *Accessible* state A, which gives $q = [1,0,0]$. In steady state, the visit count of each transient state has the following equation:

$$V_i = q_i + \sum_{j \in S_i} V_j Q_{ji} \tag{5}$$

With (4) and (5), we get
state. Therefore, the reliability of the service could be calculated as follows:

$$R_i(CS) = 1 - \sum_{i=1}^{n} (1 - R_i(t))$$

Hence, the reliability of all the services is

$$R(CS) = \prod_{j=1}^{K} R_j(CS)$$

4 FAILOVER

Failover is a backup operational mode in which the functions of a system component (such as a processor, server, network, or database, for example) are assumed by secondary system components when the primary component becomes unavailable through either failure or scheduled down time. Used to make systems more fault-tolerant, failover is typically an integral part of mission-critical systems that must be constantly available. The procedure involves automatically offloading tasks to a standby system component so that the procedure is as seamless as possible to the end user. Failover can apply to any aspect of a system: within an personal computer, for example, failover might be a mechanism to protect against a failed processor; within a network, failover can apply to any network component or system of components, such as a connection path, storage device, or Web server.

Originally, stored data was connected to servers in very basic configurations: either point-to-point or cross-coupled. In such an environment, the failure (or even maintenance) of a single server frequently made data access impossible for a large number of users until the server was back online. More recent developments, such as the storage area network (SAN), make any-to-any connectivity possible among

servers and data storage systems. In general, storage networks use many paths - each consisting of complete sets of all the components involved - between the server and the system. A failed path can result from the failure of any individual component of a path. Multiple connection paths, each with redundant components, are used to help ensure that the connection is still viable even if one (or more) paths fail. The capacity for automatic failover means that normal functions can be maintained despite the inevitable interruptions caused by problems with equipment.

5 RELIABILITY, AVAILABILITY AND SECURITY

Cloud means multiple systems acting as one. When modelling infrastructure for offering IT services to deliver reliable services is challenging and consumed may have changed with cloud computing, it is still critical for these new solutions to support the same elements that have always been important for end users. Whether the cloud serves as a test bed for developers prototyping new services and applications or it is running the latest version of a popular social gaming application, users expect it to be functioning every minute of every day. To be fully reliable and available, the cloud needs to be able to continue to operate while data remains intact in the virtual data center regardless if a failure occurs in one or more components. Additionally, since most cloud architectures deal with shared resource pools across multiple groups both internal and external, security and multi-tenancy must be integrated into every aspect of an operational architecture and process. Services need to be able to provide access to only authorized users and in this shared resource pool model the users need to be able to trust that their data and applications are secure.

A private cloud provides the same basic benefits of public cloud. These include self-service and scalability; multi-tenancy; the ability to provision machines; changing computing resources on-demand; and creating multiple machines for complex computing jobs, such as big data. Chargeback tools track computing usage, and business units pay only for the resources they use.

6 THINGS TO KNOW BEFORE MOVING TO CLOUD

The phrase "working in the cloud" used to be unfamiliar to many, but it's the way smart business is done today. Cloud computing has proven to be a secure, cost-effective way for organizations to meet accessibility, functionality, and flexibility needs in their operations. From managing document approval to running accounts payable, human resources, and order fulfilment, cloud software is not just an accepted platform, it's critical for keeping competitive in today's business environment.

When it comes to document management in the cloud, the benefits are immediate and long-lasting. From reliability, cost, and performance perspectives, there's no other option that offers the freedom and capability of cloud computing. Before partnering with a cloud provider, consider these four things in each of the following ways:

6.1 Security of Cloud Storage

Security stands as one of the main concerns for most firms with regard to cloud computing. Given this, any trustworthy cloud-based document management software will come packed with measures designed to prevent data breaches and prevent hacking.

You will want to look for:

- Multi-location data centers on multiple grids and backed up to secondary data centers in real-time to ensure business continuity
- Segregated customer data stores to create a multitenant environment without having your data shared in the same logical location with someone else's
- Independent auditors conducting regular SOC (service organization controls) audits of processes ranging from product development to data center management
- Cloud management software systems that run daily vulnerability tests performed by leading anti-virus software
- An infrastructure built on best-of-breed equipment for maximum performance and uptime
- Ongoing third-party vulnerability assessments
- Encryption capabilities to ensure that in the unlikely event that documents are accessed by an unauthorized person, s/he won't be able to view any data
- IP-based access restrictions to ensure data isn't shared in less secure environments, such as a mobile device connected via free airport or coffee shop WiFi

6.2 Reliability in the Cloud

None of the above security measures mean much if a firm's employees can't rely on cloud-based software to be

ready when they need it. Lucky for them, cloud computing is designed specifically to be available *at all times*, no matter where, when, or how people want to work. Instead of needing to rely on the infrastructure present in an office or the IT personnel on-hand, cloud software providers host software on their own servers, with open-ended availability. Better still, the software responds quickly to new asks, meaning individual companies won't have to wait for system infrastructure to scale up to meet changing, growing needs.

And most importantly, expected uptime when computing in the cloud is a very high 99.9%. To ensure this, reliable cloud software has independent monitoring systems housed from different worldwide locations constantly analyzing uptime and responsiveness. Best-of-breed cloud software will also incorporate route control technology that selects the best routing path available at any given time to guarantee optimal responsiveness.

6.3 Compliance Controls of Cloud Based Software Providers

Any cloud-based software provider worth its salt will utilize compliance controls with SSAE 16, which allows service organizations, including cloud computing providers, to disclose control activities and processes to their customers and their customers' auditors. Regulations like Sarbanes-Oxley, Gramm-Leach-Bliley, and HIPAA requirements mean that corporations have to audit their suppliers' internal controls, too, and SSAE 16 offers comprehensive audit reporting to facilitate compliance with each of these important regulations.

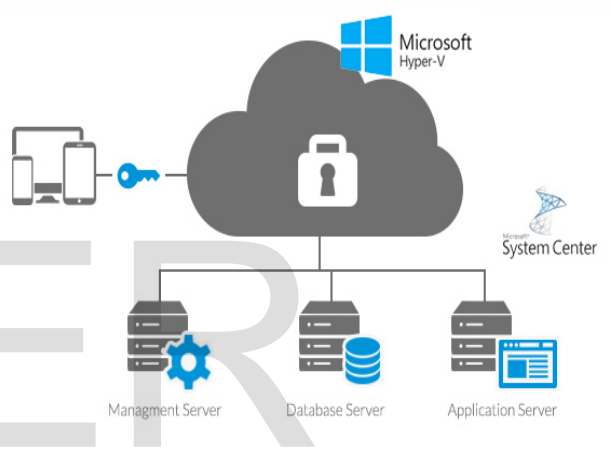
The best companies also work with data centres that have been SOC 2 Type 2 audited, which guarantees a cloud service provider can keep its clients' most sensitive data secure. Not only is it a good idea to use SOC 2 Type 2 audited companies for your cloud storage and management needs, in some cases it's increasingly required by regulatory agencies and auditors.

6.4 Creditibility of Cloud Computing Service

Anyone with a server and a software portal can claim to offer "cloud computing," but you need to be certain that you're working with a credible service provider. Insisting on only doing business with firms that meet the compliance requirements above is a great way to ensure that you're using a trustworthy source. Another good idea is investigating details like awards received and the length of time the firm has been providing their cloud computing services, as it provides a larger, clearer picture of the company as a whole. You might also want to know about their other clients: the larger and more established the client list, the more trustworthy the company.

- The infrastructure services are always activated on demand, with no upfront investment or commitment
- Resources are elastic: consumption can be increased or decreased at any time, as the situation demands.
- The infrastructure and services are fully automated in the system center and totally controlled by the cloud service provider.

Fig 1 ***Architecture of managed azure cloud***



- **Software as a Service or SaaS** is also called as AaaS – Applications as a Service. Applications are hosted by the cloud service provider (CSP), which allows the service to be accessed from anywhere at any time. The service can be any software-based application, including communications, in real-time.
- **Platform as a Service (PaaS):** A set of software and product development resources hosted on the CSP's infrastructure. Developers can create applications on the platform over the Web using APIs, website portals, or gateway software installed on the customer's computer
- **Infrastructure as a Service (IaaS)** which provides virtual server instances with unique IP addresses and blocks of storage on demand. This is sometimes referred to as "utility computing".

7 CONCLUSION AND FFUTURE WORK

Reliability has always been an important issue for design and optimization of computing systems. In service computing systems, the classical reliability model and evaluation method are no longer feasible. In this paper, we summarize the relevant researches and sources of failure on reliability of cloud computing system, and propose a a hybrid model for service availability based on the MTTF/MTTR and CTMC models for a service. Based on the model, we give the formal calculation of the reliability. We hope that this work could offer a useful reference for design and optimization of cloud computing systems. One of the goals of our future work is to design and conduct experiments based on real complex cloud computing systems.

REFERENCES

[1] IEEE guide for the use of IEEE standard dictionary of measures to produce reliable software. 1988. IEEE Std 982.1-1988, Institute of Electrical and Electronics Engineers.

[2] Reliability evaluation of cloud computing systems using hybrid methods, Intelligent Automation and Soft Computing, 2013 Vol. 19, No. 2, 165–174, http://dx.doi.org/10.1080/10798587.2013.786969.

[3] Brosch, Franz, Zimmerova, Barbora. Design-Time Reliability Prediction for Software Systems.

[4] Madan, B. B., Gogeva- Popstojanova, K., Vaidyanathan, K., & Trivedi, K. S. (2002). Modeling and quantification of security attributes of software systems [C]. Proc. International Conference on Dependable Systems and Networks, 505–604, doi10.1109/DSN.2002.1028941.

[5] V. Janardhan Babu, (Dr. C. Nadhamuni Reddy, Dr A. Govardhan) International Journal on Computer Science and Engineering (IJCSE), 3(10), **citation 3444 – 3450 Title:** Wide area networks, Monte-carlo simulation, Stochastic programming, Network Reliability. *pub id : 103-498-041,*

[6] V. Janardhan Babu, (Dr. C. Nadhamuni Reddy, Dr A. Govardhan) International Journal of Engineering and Innovative Technology (IJEIT),Volume 1, Issue 5, May 2012, *Citation 115-121.**Title:**Redundancy Design of Wan For Maximum Reliability Using Stochastic Dynamic Programming.*

[7] V. Janardhan Babu, (G. Bala gangadhar, S. Jeelan, D. Arun Prasad), International Journal of Computer Science & Communication Networks, Volume 1, Issue 3, citation 354-359. **Title:** International Journal of Computer science & Communication Networks, Improving Network Reliability Evaluation of Distance Vector Routing with Nodes reliabilities stochastic in Nature.

[8] Birolini, A. (2007). Reliability engineering theory and practice[J] (Fifth edition, pp. 2–10). Berlin: Springer

[9] Koren, I., & ManiKrishna, C. (2007). Fault-tolerant systems[M] (pp. 4–6). Morgan Kaufmann Publisher