International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 4 Issue 05, May-2015

A Review of Availability Mechanisms in Dynamic Cloud Computing Environments

Mbogholi J Msagha,
PhD Student, School of Computing & Informatics,
Maseno University,
Maseno, Kenya

Dr Henry O Okoyo,
School of Computing & Informatics
Maseno University,
Maseno, Kenya

Dr Okoth Sylvester J McOyowo,
School of Computing & Informatics,
Maseno University,
Maseno, Kenya

***Abstract*— Cloud computing is arguably a ubiquitous technology in today's digital age. The cloud has changed the way users utilize computing services such as applications and storage and thin clients is the buzzword in the fast changing consumer world of computing. Cloud Service Providers (CSPs) have been striving to achieve the magical five nines of availability, that is, 99.999% availability of the cloud. This has proven to be elusive due to outages in recent years and availability has been a major challenge to CSPs globally. Different engineers have come up with different mechanisms for increasing availability in the cloud but the numbers show that no single mechanism can increase availability effectively enough to achieve the magical five nines. This paper performs a survey of current common availability mechanisms with a view to highlighting the strength and weaknesses of each of these mechanisms and opines that perhaps engineers should now start thinking of multi-mechanism solutions in order to increase availability in the cloud.**

Keywords*— C*loud computing, availability, availability mechanisms, cloud service providers

I. INTRODUCTION

As one of the main drivers of increased uptake and consequent reliance on internet services, cloud computing has changed the face of the digisphere. Terms that are associated with the cloud include available, scalable, convenient, configurable resources and on-demand computing. Availability of the cloud has been a challenge to Cloud Service Providers (CSPs) with reliable statistics from IWGCR (International Working Group on Cloud Computing Resiliency) showing an increase in downtime steadily over the years [12]. Many other organizations monitoring availability are in agreement that in 2014 two organizations, namely Google and Amazon experienced better uptime than in previous years [14]. However, both organizations are yet to achieve the magical five nines and other CSPs are not experiencing this improvement in uptime. This paper first defines cloud computing, its benefits, architecture, and deployment models. It then examines cloud outage definition and statistics on these up to 2014. An examination and grouping of documented causes of cloud outages then follows. The paper will then describe availability in cloud computing and provide a summary of common availability mechanisms, explaining the advantages and disadvantages of each of these mechanisms. A conclusion follows that suggests the way forward in the attempt to increase availability in cloud computing.

A. Introduction to Cloud Computing

The definition of cloud computing provided by the National Institute of Standards and Technology (NIST) is a "model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (such as networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction[22]. Consequently, therefore, no organization using cloud services ought to have to worry about availability of the service or the probability of the infrastructure crashing.

As society and businesses at large become more and more dependent on the cloud for services an outage is like having a nightmare during the day for a user of the service. Usage of the cloud in today's digital age is of a stupendous nature; there are cloud communities all over the world in digisphere and statistically usage is going up and not down [3]. Arising from this one can only imagine the far reaching implications of service failure at the infrastructure level.

B. Cloud Computing Benefits, Architecture and Deployment

1) Cloud Computing Benefits

Cloud computing is the direction industry is shifting to due to the perceived overall reductions in costs that organizations will experience when they take up cloud services. Of all its advantages and benefits the strength of scalability is arguably the cloud's biggest strength; users get to use only the resources they need and they can scale up and down as the need arises, paying only for what they use. Other benefits offered by the cloud include:

- Lower total cost of ownership for organizations as they do not need to invest in infrastructure; the cloud provides this for them
- Always on, always available services
- Scalability implies no wasted capacity
- Revenue models which benefit the consumer such as pay-as-you-go and try-before-you-buy
- Economic disaster recovery solutions
- No additional software required by users
- Online deployment and development tools

The cloud computing paradigm has steadily gained momentum globally with more users enjoying the benefits of the different "IT"-as-a-service models.

2) Cloud Architecture:

The main cloud services are:

- Infrastructure-as-a-service (IaaS): this is the infrastructure at the lowest level of the model that offers a comprehensive infrastructure consisting of servers, storage, software and networks which are placed at the disposal of the user, e.g., Amazon S3
- Platform-as-a-service (PaaS): a mid-level service targeted at developers that provides them with a platform for development and deployment of applications, e.g., Microsoft Azure
- Software-as-a-service (SaaS): at the top of the stack in the model and it provides users with different software applications over the Internet that they can use, e.g., SalesForce CRM

3) Cloud Deployment Models

Additionally these services are deployed using three cloud deployment models:

- Public cloud: these are clouds provided by third parties which are available to all users who access them usually for a fee.
- Private cloud: these are clouds built by organizations in their private capacity and hosted and managed by their respective ICT departments.
- Hybrid cloud: this is a mix or hybrid of a public and private cloud model. It is the integration of on-premises IT infrastructures and internal cloud applications with applications and information deployed to a service provider (a.k.a. cloud bursting) either on a temporary or permanent basis [5].

The choice of deployment model is dependent on the needs of the enterprise.

II. CLOUD OUTAGES

SearchCloudStorage define a cloud outage as "a period of time during which cloud services are unavailable." [16]. As time has gone by the number of cloud outages at service provider level has gone up. The Cloud Security Alliance (CSA) Cloud Vulnerabilities Working group pointed out that the number of cloud vulnerability incidents doubled between 2009 and 2011[7]. More recent data collection by The International Working Group on Cloud Computing Resiliency (IWGCR) shows an increase of 70% in reported downtime across the service providers sampled from 240 hours to 410 hours [9]. Further in their latest report covering 2013 statistics from the same service providers show an whooping increase of 218% in downtime from 410 hours in 2012 to 1305.21 hours in 2013 [7]. This is definitely a worrying trend all factors notwithstanding. It is exacerbated further by the fact that most service providers do not disclose the cause of these outages; in fact, the CSA Cloud Vulnerabilities group observed that 25% of reported outages do not disclose the cause of the outages. However, the same group opined that most service providers had begun reporting outages since 2010 nevertheless.

To give an idea what the cost of outages is to business consider the following facts:

- Amazon's outage of January 2013 which lasted 49 minutes cost them upwards of US $ 4 million in sales. In August of the same year a 30 minute outage cost them an estimated US $ 66240 per minute [23].
- Google's 5 minute outage in 2013 cost them an estimated US $ 500000. This issue alone led to a 40% drop in Internet traffic worldwide [23].

A. *Causes of Cloud Outages*

There are various causes of cloud outages and numerous inherent factors that cause risk of the cloud infrastructure failing. Though it has been noted in a previous section that the CSA (Cloud Security Alliance) Cloud Vulnerabilities group had observed more opening up by CSPs on causes of outages since 2010 nevertheless there is still reason to have more disclosure from these players; the reason for this will be apparent in later sections of this review. However, some of the risks and causes of cloud outages are examined next.

Myerson [13] noted the following as the types of failures that trigger outages:

- Leap year failure
- Numerically unstable algorithms
- Resource Optimization failure
- Threshold policy implementation failure
- Hypervisor failure
- Virtual desktop failure

Between 2009 and 2012 the CSA proposed a list of the highest category of threats to cloud computing. These threats together with the work of Li et al. [11] and Potharaju [15] provide different causes of outages both on and in the cloud. For the purposes of this study these outages can be grouped together as:

- Resource exhaustion, node failures, network issues, natural disasters, security issues, configuration issues and hardware issues.

With all these outages at the different levels in the cloud how have service providers been building for availability?

III. Increasing Availability and Fault Tolerance In the Cloud

Availability in cloud computing is generally measured using the 9's measurement with the count of 9s being a percentage, e.g. three 9's implies (99.9%) uptime while four 9's implies (99.99%) uptime and so on. This is normally specified in the service provider's SLA (Service Level Agreement) with the ideal uptime environment being the five 9's (99.999%) uptime also referred to as the magical five nines in the computing industry. The uptime percentages are based on annual availability minus any time it takes for maintenance and scheduled outages. The cost of downtime has been discussed in section II and it goes without saying that higher uptime (read more 9's) is more desirable. As an indicator three 9's means 99.9% uptime which implies nine hours of downtime per year (outside the scheduled outages and maintenance outages) while four 9's means 53 minutes of downtime per year.

The foregoing interpretations of the term availability imply that an available system (or network for that matter) is one that is accessible, ready to use at any given time and information or resources on it can be accessed and used in the correct format. Suffice to say this is what cloud service providers promise to users of their services and in return users expects no less. Cloud service providers have used different techniques to ensure availability to their users and it is worthwhile noting that availability must be addressed at both datacenter as well as at infrastructure level.

Most service providers build for fault tolerance in their infrastructure. Fault tolerance is generally defined as the ability of a system to remain in operation even if some of the components used to build the system fail [1].

A summary of the current Availability Mechanisms (AMs) is provided below:

1) Replication: Hauck et al [9]

Mechanism: By replicating servers and storage across network. The authors describe this technique as "the key technique to achieve high availability". The technique works in such a way that each server has a replica that will take over from it in the event that it fails; thus, in a network with ten servers there will be a minimum of ten replicas thus increasing the overall number of servers on the network to at least twenty. The same would apply to storage, with each storage device having a replica that would also take over in the event that the primary storage fails.

- Advantages: When one server goes down another one takes over; same with storage
- Disadvantages: the CAP principle [6][8]. Have to make a choice between transactional consistency (C), high availability (A), and resiliency to network partitions (P). This effectively means that it is not possible to achieve consistency, replication and high availability all at the same time, in a typical cloud environment. The implication is that high availability can be achieved with replication or consistency across the network, but not both.

2) Checkpointing: Singh et al.[17]

- Mechanism: The authors proposed that availability can be increased by improving checkpoint efficiency and preventing check pointing from being the bottleneck of cloud data centers. A checkpoint is a local state of a job saved on stable storage. Checkpoints work like restore points for an operating system such that the status of a process can be saved at consistent intervals so that if there is failure computation can be resumed from the earlier checkpoints, thereby avoiding restarting execution of the job from the beginning again. When a node fails at either the service manager or service node level, the threads can be re-allocated to other nodes which will take up the execution since in cloud computing environments nodes in the data centre do not share memory. They went on further to examine the check pointing scheme using two main metrics: checkpoint overhead (increase in the execution time of the job because of a checkpoint implementation) and checkpoint latency (duration of time required to save the checkpoint). By performing a multilevel checkpoint analysis in a simulated environment they observed the shortcomings of [26] and [4]. Essentially by varying the checkpoint rerun time [17] proposed two load balancing algorithms to cater for the multilevel proposition so that execution time for a job could be minimized. Consequently by improving checkpoint efficiency this would prevent checkpointing from being bottleneck in the cloud centre.
- Advantages: as checkpoints act like restore points of an OS, they reduce job execution total time in event of node failure at manager or node level by re-allocating threads in the shared-nothing environment (uses checkpoint overhead and checkpoint latency as metrics)
- Disadvantages: Based on rollback recovery which is reactive not proactive; does not address a particular outage cause since some outages affect the nodes themselves making this mechanism ineffective in such instances

3) HA-OSCAR: Thanakornworakij et al. [21].

Mechanism: High Availability Open Source Cluster Application Resource. This mechanism uses redundancy at all levels. OSCAR is a cluster software stack that provides a high performance computing runtime stack and tools for cluster computing [2]. Cluster computing is whereby more than one computer is connected together to act and appear as one computer. The main goal of the HA-OSCAR project was to leverage existing OSCAR technology, so the HA-OSCAR project was formed to provide high-availability capabilities in OSCAR clusters. HA-OSCAR then introduced several enhancements and new features to OSCAR mainly in areas of availability, scalability and security. The proposed system (HA-OSCAR 2.0) would use the concept of component redundancy to eliminate single-point-of-failures. It would utilize HATCI (High Availability Tools Configuration and Installation). HATCI is composed of three components: Node Redundancy, Service Redundancy and Data Replication Services. As seen in figure 1 below there is redundancy from the head node down to the switches and to the client nodes thus ensuring if any primary device fails then a secondary device can take over its place. The head node provides service requests from users and routes appropriate tasks to the

compute nodes (essentially the request manager described by [17] previously). An evaluation was then performed to demonstrate improved availability in an OSCAR-V and HA-OSCAR integrated environment. By performing an availability and cluster system analysis the authors reached the conclusion that availability for OSCAR-V cluster system was 0.996 while it was 0.99999 for the HA-OSCAR V cluster system. This translates to a downtime of 39.2 hours and 4.25 minutes annually respectively.

- Advantages: Efficient failover mechanisms at all levels; works well in cluster environments
- Disadvantages: designed for cluster environments; does not address a particular outage cause, e.g. could failure of primary device cause failure of failover device?

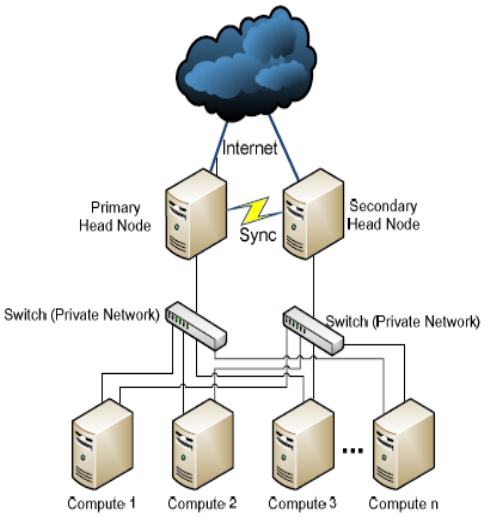

Fig. 1. A typical HA-OSCAR V Cluster System[19]

4) Linux-HA: Linux-HA.org.

Mechanism: It uses Heartbeart software in a passive-active environment which detects node failures in less than half a second. The project's main software product is Heartbeat, a GPL (General Public License)-licensed portable cluster management program for high availability. Heartbeat can detect node failures reliably in less than half a second. With a low latency communication infrastructure, such as Infiniband or Myrinet, this time could be lowered significantly. The architecture is based on an active-passive high availability solution (where one server is active while the other one remains passive until the active one fails, in which case it takes over). Each service under high availability needs at least two identical servers: a primary host, in which the service run, one or more secondary hosts, able to recover the application in less than one second. As a result of failure detection, the active-passive roles are switched. The same procedure can be done manually, for planned or unplanned down time, i.e. in case of maintenance needs. A heartbeat keep-alive system is used to monitor the health of the nodes in the cluster. Heartbeat monitors node health through communication media, usually serial and Ethernet links. It is a good solution to have multiple redundant connection links. Each node runs a heartbeat daemon process. When a node death is detected, Heartbeat runs a script to start or stop services on the secondary node. A local disaster recovery solution is typically composed of two homogeneous nodes, one active and one passive. The active node is usually called master or production node, and the passive node is called secondary or standby node. During normal operation, the only working node is the master node; in the event of a node failover or switchover, the standby node takes over the production role, by taking its IP number, and completely replacing the master one. To maintain the standby node for failover, the standby node contains homogeneous installations and applications: data and configurations must also be constantly synchronized with the master node.

- Advantages: very versatile and relies on Heartbeat for detection
- Disadvantage: ideally designed for specific environments, namely Linux, FreeBSD, OpenBSD, Solaris and Mac OS X

5) CloudDisco: Stanik, Hoger and Kao [18].

Mechanism: Cloud managers act as peers. This solution offers availability at the cloud middleware level using a self-healing mechanism incase of failure at any of the three points depicted in figure 2 below:

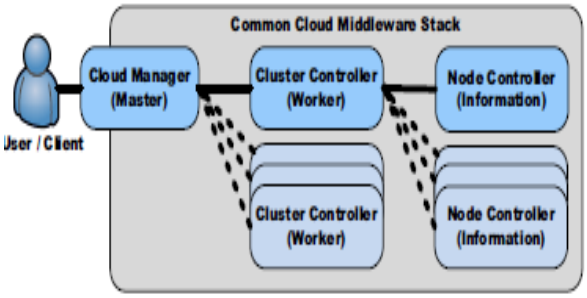

Fig. 2. Middleware components in 3-tiered master-worker architecture [16]

In the architecture above the cloud manager acts as a master for the cloud environment receiving requests directly from the user for Infrastructure (hardware). The cloud manager in turn passes this request to the cluster controllers (workers). The controller workers process the request and pass it back to the cloud manager who in turn also notifies the user. It can be observed in this architecture if the cloud manager fails then the entire cloud fails even if both the cluster controllers and node controllers are available. CloudDisco offers a multi-master architecture which the authors claim not only prevents failure but also offers a self-healing mechanism in the event of failure in any of the levels of the architecture.

In the multi-master architecture all cloud managers are peers and act as a unit; thus they are not replicas but active peers which collectively make up the cloud, i.e. each cloud manager owns a fraction of the cloud. All cluster controllers are connected to exactly one cloud manager at any given time and since all cloud managers are peers, users and cluster controllers alike, can connect to any cloud manager. The cluster controllers do not have any knowledge of each other

(operating like a shared-nothing environment) but are connected to at least one node controller (resource provider). In the above architecture each cloud manager must be connected to at least one other cloud manager and the collective collection (of cloud managers) results in a mesh topology between cloud managers and a tree topology with each master architecture. In the event of failure of a cloud manager the cluster controller within it can connect to another cloud manager by means of a mechanism thus ensuring no total cloud blackout and a better failover mechanism. It is also worth mentioning that the authors bore scalability in mind when developing this architecture, for the most part through the cloud managers. There are many master nodes available in this setup and these ensure scalability since user requests can be distributed across all available master nodes (cloud managers).

- Advantages: Versatile; has self-healing mechanism
- Disadvantages: could a multi-attack bring reliability issues through bottlenecking and even cause resource exhaustion?

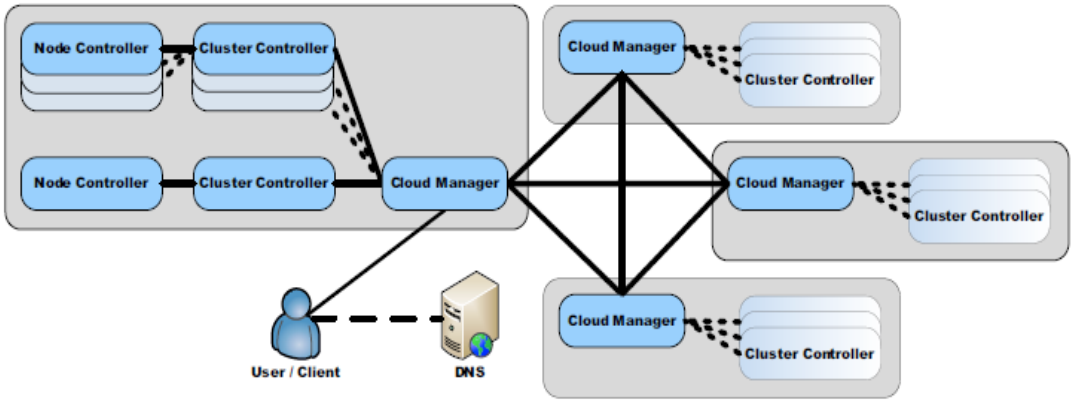

Fig. 3. CloudDisco multi-master architecture layout [16]

Other AMs have been developed by the following:

- Proxy Network: [25].
- Using specialized middleware: [10].
- Software Defined Availability: [20].
- Vmware HA solution: [24].
- Collaborative Fault Tolerance: [19].

IV. CONCLUSION

It is literally impossible to separate availability from outages; the two are like Siamese twins with availability being the good twin while outage is the evil twin. The review presented above has examined the different documented causes of outages at cloud level. It has also been shown how outages affect businesses and users. The different authors cited in part II have shown the severity of outages to businesses, and they have investigated the causes of these outages using different means. The CSA has gone further to even show the severity of some disruptions as well as given examples of such disruptions.

Part III has discussed the different ways in which researchers and industry players have been building for high availability in the cloud. The main mechanisms observed are the use of redundant mechanisms, virtualization, use of cluster computing and software defined availability. A self-healing mechanism was designed by [18].

From the two sections it appears there is a disjoint between the two groups of authors (those studying outage causes and those offering availability solutions). We suggest as a future work that there is a way that the two groups could combine their research and findings in a way that will allow them to understand the twins with a view to increasing availability in the cloud. Further we believe that subsequent to this, a multi-mechanism solution approach may prove to be the way for the future of enhancing availability in dynamic cloud computing environments.

REFERENCES

[1] Amazon Web Services (2010). Building Fault-Tolerant Applications on AWS. Retrieved from http://d36cz9buwru1tt.cloudfront.net/AWS_Building_Fault_Tolerant_Applications.pdf

[2] Brim, M.J., Mattson, T.G., Scott, S.L.(2001): OSCAR: Open Source Cluster Application Resources. In: Ottawa Linux Symposium 2001, Ottawa, Canada (2001)

[3] Businesswire.com (2013) 2013 Future of Cloud Computing Survey Reveals Business Driving Cloud Adoption in Everything as a Service Era; IT Investing Heavily to Catch up and Support Consumers Graduating from BYOD to BYOC (Reported on June 19, 2013)

[4] Daly, J. (2006). A higher order estimate of the optimum checkpoint interval for restart dumps,‖ Future Generation Computer Systems, pp 303-312, 2006

[5] Emerson Network Power white paper (2010). Taking the Enterprise Data Center into the Cloud: Achieving a Flexible, High-Availability Cloud Computing Infrastructure. Retrieved from http://www.EmersonNetworkPower.com

[6] Fox, A. and Brewer, E.A. (1999). Harvest, yield, and scalable tolerant systems. In HOTOS '99: Proceedings of the Seventh Workshop on Hot Topics in Operating Systems, page 174, Washington, DC, USA, 1999. IEEE Computer Society

[7] Gagnaire, M., Diaz, F., Coti, C., and Cerin, C. (2012). Downtime statistics of current cloud solutions. … Working Group on Cloud …, 2–3. Retrieved from http://iwgcr.org/wp-content/uploads/2012/06/IWGCR-Paris.Ranking-002-en.pdf

[8] Gilbert, S. and Lynch, N. (2002). Brewer's conjecture and the feasibility of consistent available partition- tolerant web services. In ACM SIGACT News, p.2002

[9] Hauck, M., Huber, M., Klems, M., Kounev, S., Quade, J.M., Pretschner, A., Reussner, R., Tai, S.(2010). Challenges and Opportunities of Cloud Computing: Trade-off Decisions in Cloud Computing Architecture. In Karlsruhe Insitute of Technology Technical Report Vol. 2010-09

[10] Kanso, A., and Lemieux, Y. (2013). Achieving High Availability at the Application Level in the Cloud. 2013 IEEE Sixth International Conference on Cloud Computing, 778–785. doi:10.1109/CLOUD.2013.24

[11] Li, Z., Liang, M., Brien, L. O., and Zhang, H. (2013). The Cloud ' s Cloudy Moment : A Systematic Survey of Public Cloud Service Outage. International Journal of Cloud Computing and Service Science, Vol 2, No5, pp 321-331

[12] Maurice Gagnaire (France), Felipe Diaz (Colombia), Camille Coti (France), Christophe C´erin (France), Kazuhiko Shiozaki (Japan), Yingjie Xu (China), Pierre Delort (France), Jean-Paul Smets (France), Jonathan Le Lous (France), Stephen Lubiarz (France), Pierrick Leclerc (France). Downtime statistics of current cloud solutions, IWGCR Report 2013 & Update 2014

[13] Myerson, J. M. (2013). Mitigate risks of cloud resource exhaustion outages Use service level agreements and other proactive tools to avoid, IBM developerworks, 1–9.

[14] Network world.com (2015) Which cloud providers had the best uptime last year? Url: http://www.networkworld.com/article/2866950/cloud-computing/which-cloud-providers-had-the-best-uptime-last-year.html (accessed 8th May 2015)

[15] Potharaju, R. (2011). When the Network Crumbles: An Empirical Study of Cloud Network Failures and their Impact on Services. Microsoft Research, Redmond

[16] Searchcloudstorage.com(2015) http://searchcloudstorage.techtarget.com/

[17] Singh, D., Singh, J., and Chhabra, A. (2012). Evaluating Overheads of Integrated Multilevel Checkpointing Algorithms in Cloud Computing Environment. International Journal of Computer Network and Information Security, 4(5), 29–38. doi:10.5815/ijcnis.2012.05.04

[18] Stanik, A., Hoger, M., and Kao, O. (2013). Failover Pattern with a Self-Healing Mechanism for High Availability Cloud Solutions. 2013 International Conference on Cloud Computing and Big Data, 23–29. doi:10.1109/CLOUDCOM-ASIA.2013.63

[19] Tchana, A., Broto, L., and Hagimont, D. (2012). Fault Tolerant Approaches in Cloud Computing Infrastructures. ICAS 2012, The Eighth International Conference on Autonomic and Autonomous Systems pp 42-48

[20] Teich, P. (2014). Software Defined Availability (SDA): Critical for Managing Datacenter Scale (pp. 1–9). Technical Report by Moor Insights and Strategy

[21] Thanakornworakij, T., and Sharma, R. Blaine S, Chokchai L, Zeno G, Pierre R,Christine M.(2012). High availability on cloud with HA-OSCAR. Euro-Par 2011: Parallel …, 292–301. Retrieved from http://link.springer.com/chapter/10.1007/978-3-642-29740-3_33

[22] The National Institute of Standards and Technology (NIST), Information Technology Laboratory definition of Cloud Computing by Peter Mell and Tim Grance, version 15, October 7, 2009.

[23] Vmblog.com (2013). The Outrageous Cost of Downtime. http://vmblog.com/archive/2013/09/10/infographic-the-outrageous-costs-of-data-center-downtime.aspx#.U_hrfle03Eo, accessed 11/08/14

[24] VmWare. (2014). VMware High Availability; Concepts, Implementation and Best Practices. Technical Report by VMWare.

[25] Weissman, J., and Ramakrishnan, S. (2009). Using proxies to accelerate cloud applications. … of the Workshop on Hot Topics in Cloud …. Retrieved from https://www.usenix.org/event/hotcloud09/tech/full_papers/weissman.pdf

[26] Young,J.W.(1974). A first order approximation to the optimum checkpoint interval,‖ Communications of the ACM, 17(9):530531, 1974