This is the published version of a Proceedings Paper presented at the **2012 15th IEEE International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing Workshops**

Garraghan, P, Townend, P and Xu, J (2012) *Real-time fault-tolerance in federated cloud environments.* In: Proceedings - 2012 15th IEEE International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing Workshops, ISORCW 2012. UNSPECIFIED. IEEE , 118 - 123. ISBN 978-1-4673-0900-4

2012 IEEE 15th International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing Workshops

Real-Time Fault-Tolerance in Federated Cloud Environments

Peter Garraghan, Paul Townend, Jie Xu
{scpmg, p.m.townend, j.xu} @ leeds.ac.uk
School of Computing, University of Leeds, LS2 9JT, UK

***Abstract*—Dependability is a critical concern in provisioning services in Cloud Computing environments. This is true when considering reliability, an attribute of dependability that is a critical and challenging problem in a Cloud context [2]. Fault-tolerance is one means to attain reliability, and is typically implemented by using some form of diversity. Federated Cloud, which is an emerging Cloud paradigm that orchestrates multiple Clouds, is able to implement environmental diversity for Cloud applications with relative ease and minimal additional cost to the consumer due to its inherent design. Real-Time Applications (RTAs) can benefit from deploying fault-tolerant schemes to fulfill deadlines in the presence of faults as they enable the provisioning of correct service in the event of a component in the application failing. However, this diversity can potentially become an issue when designing dynamically scalable fault-tolerant RTAs in a federated Cloud environment while also fulfilling QoS demands. In particular, building fault-tolerant RTAs by using the diversity of the Virtual Machine (VM) configurations and of the underlying Cloud infrastructure can have a negative impact on the ability to fulfill deadlines whilst still allowing the application to dynamically provision VMs with minimal human interaction. This paper identifies a number of characteristics that affect the ability for a RTA to fulfill specified deadlines in a federated Cloud environment as a result of deploying environment diverse fault-tolerant schemes. Furthermore we have designed and performed initial experiments using a real world Cloud federation to justify the feasibility of this problem. Results demonstrate that deploying RTAs in a federated Cloud environment can potentially increase the rate of deadline violations.**

***Index Terms*— Cloud Computing, Cloud Federation, Real-Time Cloud Computing, Dependability, Fault-Tolerance.**

I. INTRODUCTION

Cloud computing has emerged as a computing paradigm to enable the provisioning of resources by the use of large scale computing infrastructure. It provides the ability to dynamically scale-up and scale-down resources in accordance with the requirements of the users of that infrastructure. As defined by the National Institute of Standards and Technology (NIST) [1], Cloud computing is "a model for enabling convenient, on demand-network access to a shared pool of configurable resources that can be provisioned and released with minimal management effort or service provider interaction". Clouds are typically characterized as large-scale systems that leverage virtualisation technology to support multi-tenancy. Real-Time Applications (RTAs) require large amounts of computing resources as well as the ability to dynamically scale the required amount of resources to fulfill time deadlines and expected Quality of Service (QoS).

Cloud computing is still an evolving paradigm. As the usage and demand for Cloud grows, there are increasing limits on how much resources can be scaled and provisioned by a single Cloud [3]. A proposed solution to this problem is to federate multiple Clouds together in order to extend the scalability of Cloud systems and enable QoS to be satisfied in the face of increased consumer demand and usage of the infrastructure [4].

Writing in [17], Randell defines dependability as "that property of a computer system such that reliance can be justifiably be placed on the services it delivers. The services delivered by a system is its behavior as it its perceived by its user". Dependability is a critical concept in Cloud computing systems, as there are great economic consequences in the ability to provision dependable services [5]. Due to the large scale, complexity and multi-tenant nature of a Cloud, these systems experience an increase in the number of hardware and software faults within their respective systems to a point where fault occurrence becomes common rather than an exception. Furthermore reliability, an attribute of dependability that is defined as the continuity of correct service [22], is not clearly understood and remains a critical research problem within a Cloud context [2]. Fault-tolerance is one means to attain reliability. RTAs can benefit from deploying fault-tolerance to fulfill deadlines in the presence of system faults, as it enables the application to provision correct service in the event of failures. Reliability is also critical in RTAs, since failure of a real-time system can cause both economical disaster and loss of human lives [6]. Fault-tolerant schemes implemented in a Cloud infrastructure greatly benefit from environmental diversity, which is based on the premise that different operating environments may cause a system to follow different execution paths therefore not activating the same faults. However this same diversity can potentially become an issue when designing scalable fault-tolerant RTAs that require to be large-scale, dynamically provisioned and require minimal human interaction as a result of the environment affecting the execution time of RTAs.

This paper discusses the challenges of developing fault-tolerant RTAs as a result of environment diversity in federated Cloud architectures. Two characteristics that have a notable effect on Cloud environment diversity are identified and discussed. Furthermore, we have designed and performed

978-0-7695-4669-8/12 $26.00 © 2012 IEEE
DOI 10.1109/ISORCW.2012.30

experiments using a real world Cloud federation to justify the feasibility of this problem in relation to these two characteristics, and have identified further work as this research matures.

The remaining sections of this paper are structured as follows: Section II provides a background on Cloud computing and dependability. Section III describes the related work in the area. Section IV describes the experiment setup. Section V presents and discusses the results of the experiment. Finally, Section VI presents the conclusion and future work.

II. BACKGROUND

A. *Federated Cloud*

Cloud computing is an emerging computing paradigm to facilitate the establishment of large scale, flexible computing infrastructures that are available on demand. As the usage and storage capacity of Clouds grows, there are increasing limits on the resource capacity and scalability a single Cloud provider can provider while also meeting QoS demands [3]. A proposed solution is to federate multiple Clouds together, in order to extend the scalability of the Cloud systems, as well offer a number of additional benefits such as increase interoperability across different administration domains, improved economies of scale and removal of Cloud isolation. A number of Cloud federation models have emerged recently to address challenges such as enabling additional scalability, resource management and dependable storage [15][16]. These models are important in developing frameworks to realise the vision of Cloud federation.

B. *Dependability in Cloud*

Dependability in Clouds is a key concern, as there are potentially great economic consequences for any failures [5]. Additionally, according to [2] building reliable and available Clouds is a "critical, challenging and urgently-required research problem". There have been a number of papers discussing the differences between Cloud and other distributed systems, most notably Grid, as well as the new research challenges that Cloud brings [20]. Clouds have a large number of different deployment and service models with each of these models and deployments being composed of different architectural implementations. This broad definition of what a Cloud system actually is can make it somewhat confusing to see the justification for novel dependability work for Cloud. Cloud computing contains the following characteristics that can be found individually in other distributed paradigms, however combined together introduce important Cloud dependability concerns. These characteristics are summarised in table I.

Massively scalable and complex architectures. The use of virtualisation enables large scale systems that are able to massively scale machines to be provisioned on-demand. It is well researched that the increase in size of a system results in more complex management and architectures. As stated in section I, system size and complexity results in more frequent faults. This characteristic in such a system increase the number of hardware and software faults to a point where fault occurrence becomes common rather than an exception.

Many different consumers with different business objectives sharing the same infrastructure. Clouds typically are of a multi-tenant nature, with each consumer pursuing different business objectives. These different business objectives typically result in different dependability requirements, translating into numerous high QoS expectations depending on the consumer. Consumers expect Cloud infrastructure to have service that is equivalent and sometimes greater than their own infrastructure. A problem with this is that there are situations where there are conflicting QoS concerns as result of the multi-tenancy in Cloud.

Wide diversity of dynamic workloads. Cloud workloads are somewhat more general than those of Grid, which are characterised as being more specific to scientific batch processing. Clouds also experience dynamic workload size

TABLE I. CLOUD CHARACTERISTIS AND RELATED DEPENDABILITY CONCERNS

Cloud Characteristic	**Dependability Concern**
Massive scale and complex architectures	Increase in size and complexity of system results in more frequent failures.
Multi-tenant and high consumer QoS expectation	Conflicting consumer QoS concerns.
Diverse workload dynamicity	Cloud scalability and generality of workloads require the Cloud to handle different types of faults and failures as a result of different workloads cause different types of faults to activate.
Rapid demand and growth	Above Cloud characteristics and dependability concerns are made more crucial.

and variety, from a result of the Clouds ability to quickly scale to a consumer's needs as well as the Cloud's multi-tenancy. It has been established that different workload properties such as computationally and I/O intensive workloads influence the type of faults that are activated [18].

Demand is rapidly growing. Cloud has been reported as a rapidly growing market, which results in the above characteristics becoming a growing concern.

The problems defined are also extended in a federated Cloud environment, in particular the size of the architectures as well as the diversity of workloads. This results in a federated Cloud being able to provision more resources to more consumers than that of a single Cloud, perpetuating these problems.

One means of attaining dependability is through fault-tolerance. The effectiveness of many fault-tolerant protocols relies on the failure independence between nodes. This failure independence can be increased by the use of environment diversity, by executing the same software in a different environment so that the program follows a different execution path and does not activate the same faults [19]. Examples of variables that affect the execution environment include different processor speeds and Operating System (OS) architectures. Cloud federation by its inherit nature is environmentally diverse as a result of the diversity of administrations, architectures, geographical location, hardware, etc. which enables the development of failure independent applications in the federated Cloud at little additional cost to the consumer [7].

C. Real-Time Cloud

According to [8], Real-Time computing is defined as "the correctness of the system depends not only on the logical result of the computation but also on the time at which the results are produced". Failing to fulfill these requirements can lead to economic consequence or loss of human life. Real-Time Systems are used in a number of domains such as military, flight-control systems and financial applications. RTAs can be classified in terms of hard, firm or soft [6]. Hard RTAs are applications whose non-fulfillment of deadlines leads to service failure. Firm RTAs are deadlines where hard deadlines are enforced, however non-fulfilled deadlines does not result in service failure. Soft RTAs non-fulfillment of deadlines causes performance degradation of the system but does not cause service failure.

Fault-tolerant RTAs can leverage the use of environment diversity to increase the reliability of the application. However in a federated Cloud context this can lead to violations of timed deadlines. For example it is well established that processing power has a strong influence on the time of execution for an application. In Infrastructure-as-a-Service (IaaS) Cloud, providers provision VMs with a specified amount of computer resources such as storage, memory and computation. However when IaaS Cloud providers discuss the precise amount of processing power assigned to a VM, they are given as approximations. For example Amazon EC2 states that in their infrastructure 1 Compute Unit is the equivalent processing power of a 1.0-1.2 GHz processor [9]. RackSpace states that it is roughly the equivalent of 2.8 GHz; exactly how approximate this figure given is remains unknown [10]. Other examples of variables within the Cloud that have noticeable effect on environment diversity of a fault-tolerant RTA includes the OS the application is deployed on and the network topology between nodes.

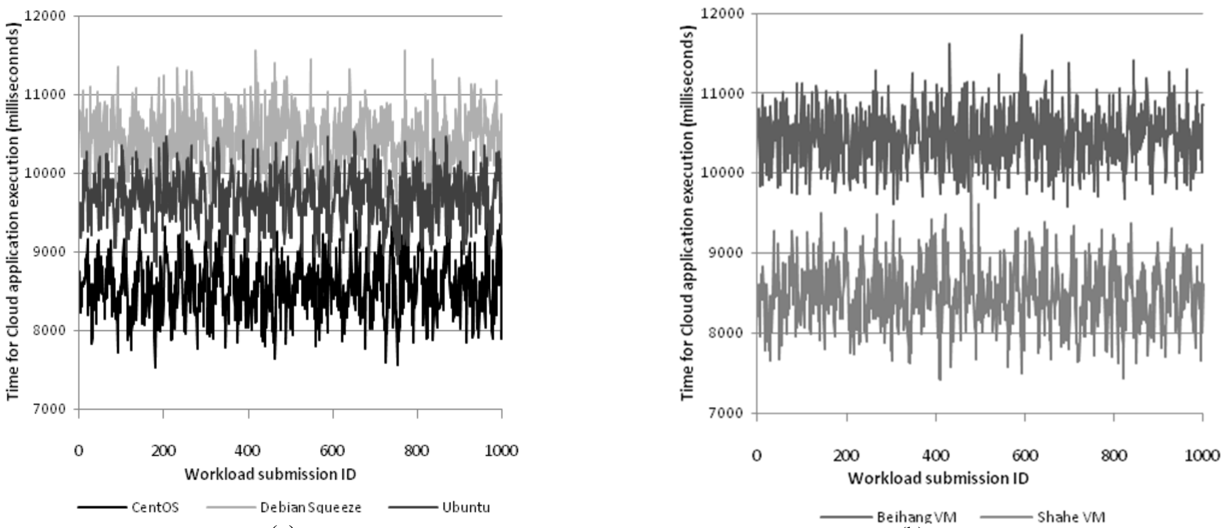
III. Related Work

There are a number of papers that focus on addressing the challenges of adopting real-time Cloud. [11] explores the challenges and needs of Cloud infrastructures for data-intensive applications requiring strong timing guarantees by adapting multiprocessor scheduling techniques for the Cloud environment. The paper concludes that one of the ongoing challenges in this research is Cloud unpredictability which decreases the effectiveness of any real-time scheduler.

[12] addresses a number of issues in enabling real-time dynamism in Cloud Computing architecture. Two points of relevance in this work is the issue of human system administrators restrict the enablement of real-time dynamism, and that policy-based management is not an optimal or scalable solution for Cloud environments due to fluctuating workload demands. This paper additionally provides support that environment diversity of the Cloud environment also strengthens the importance of these issues.

[13] identifies the issue of environment diversity in relation to power efficiency of a single heterogeneous datacenter and propose analytical predictive layers to map workloads to best fitting platforms. In the future work section they propose managing the allocation of VMs based on performance demands such as time characteristics, an area that is of great relevance.

Our approach focuses on the dependability of fault-tolerant RTAs. The motivation for this research is to highlight the impact of environment diversity in fulfilling time deadlines for RTAs that use fault-tolerant schemes in a federated Cloud environment. This problem is not only restricted to fault-tolerant RTAs but also includes deploying RTAs in a federated Cloud environment. We wish to highlight that leveraging environment diversity fault-tolerance for RTAs in a federated Cloud results in additional complexity in predicting the probability of fulfilling time deadlines. Our work differs from that of the literature by focusing on provisioning dependable RTAs and analysing a Cloud federation composed of multiple Cloud datacenters under different administrations.



(a) (b)

Figure 1. Application execution time comparison: (a) Single Cloud, (b) Federated Cloud

IV. Experiment Setup

In this paper we hypothesise that the environmental diversity across a federated Cloud environment has a noticeable impact on a Real-Time fault-tolerant application's ability to fulfill time deadlines. For our initial work, we have set up an experiment to demonstrate that deploying RTAs across a federated Cloud environment results in noticeable unpredictability in fulfilling time deadlines. To demonstrate this, we use an N-Copy scheme across a heterogeneous federated Cloud. N-Copy is similar to N-Version design systems but instead of design diversity, implements data diversity [21].

The experiment test bed is composed of two real world Clouds; Beihang Cloud and Shahe Cloud. Beihang Cloud is formed using 32 x HP BL460C blade servers, with each server consisting of 2 x Intel E5405 2.0GHz CPU and 16GB memory. Shahe Cloud is composed of 32 x Dell 6100 machines, with each machine containing 2 x Intel Xeon E5620 2.4GHz CPU and 48GB memory. Replicas are deployed in separate VMs, with the application deployed in one of three possible OSs; Debian Squeeze, Fedora and CentOS. The application deployed on each replica is based on the MoSeS e-Social Science project [14], a CPU intensive program that generates a virtual representation of a population and then performs various analyses on that population. The application deployed on each VM is N-Copy, which is acceptable in this experiment setup for two reasons. First, this enables scalability of the application by the creation of additional VMs with minimal amount of additional configuration to the application and OS; one of the main advantages of deploying scalable applications in a Cloud environment. Second, this paper is to demonstrate the difference in application execution times of RTAs as a result of environment diversity in a federated Cloud, not on the effectiveness of the fault-tolerant scheme we have deployed.

To demonstrate the differences in application completion time between VMs deployed in a federated Cloud environment, each application within a replica processes the same size dataset. A job scheduler submits a workload to each replica to begin application execution. After the application completes, the VM records the time period for completion time for the job. This completion time is server side, and does not consider network latency. The job scheduler submits the workload to the N-Copy fault-tolerant scheme a total of 1000 iterations.

V. Result Analysis

Figure 1(a) represents three VMs deployed using heterogeneous OSs within the Beihang Cloud. The figure demonstrates that the VM using CentOS has the lowest execution time, while Debian Squeeze experiences the highest execution time on average with an execution time of approximately 2 seconds greater than that of CentOS. Figure 1(a) demonstrates that the OS architecture has a noticeable impact on the application completion time even in homogeneous hardware within a Cloud environment.

Figure 1(b) represents two VMs both deployed using Debian Squeeze residing in separate Cloud infrastructures. This demonstrates that in almost all workload submissions the VM deployed in the Shahe Cloud has a lower application execution time than that of Beihang Cloud. The reason for this being that the hardware in the Shahe Cloud infrastructure has more processing power than that of Beihang Cloud, and the application itself is computationally heavy.

The results represented in figure 1(a) and 1(b) are not necessarily surprising. However what is of interest is that the hardware heterogeneity in the federated Cloud application environment when deploying identical OSs in some cases results in a sizable difference in total application execution time being comparable to that of deploying Cloud applications using heterogeneous OSs within the same Cloud infrastructure. This is an issue in RTAs for a number of reasons.

First, as identified in section II, Cloud businesses provision Compute Units as an estimation as opposed to a precise figure. In Amazon EC2's case between 1.0GHz and 1.2GHz. As demonstrated in our experiments, a difference between 2.0GHz and 2.4GHz can result in a noticeable difference in execution time, which affects the ability for a RTAs to fulfill time deadlines. This is not necessarily a problem when the application developer is required to manually create and configure each VM, allowing the developer to enter a testing phase and discover the best and worst case execution time of a replica fulfilling a deadline. However this is not feasible when a potentially large number of VMs across a heterogeneous, federated Cloud environment require to be managed without large amounts of human interaction. For example, dynamically creating more VMs to meet resource and deadline demands of the RTA application or restarting and migrating a VM as a result of a failed replica in a fault-tolerant application. As identified in [12], human interaction and configuration leads to the inability to dynamically scale, create and configure VMs in a Cloud environment which weakens one of the defining characteristics of Cloud computing.

Secondly, this heterogeneity can lead to a restriction of environmental diversity in fault-tolerant applications to meet deadlines, should a developer instead decide to deploy all VMs within the same physical infrastructure, or use the same OS to attempt avoid deadline violations. As a result, this damages the dependability of the system due to the decrease in failure dependence between nodes which causes the increased probability of common-mode failure occurring.

VI. Conclusions and Future Work

Dependability is a fundamental concern in designing dependable Cloud computing environments. Leveraging the environmental diversity of a RTA within a federated Cloud environment can improve the effectiveness of fault-tolerant schemes. However, this same diversity can have a negative impact on an application to fulfill specified deadlines. This problem can be mitigated with the manual configuration and deployment of VMs, however this diminishes the effectiveness of the Cloud environment in developing a fault-tolerant RTA as it hampers the ability to dynamically scale VMs deployed that are environmentally diverse with minimal human interaction.

This paper analyses the problem of Cloud federation unpredictability resulting in the decreased effectiveness of fault-tolerant RTAs, as well as identifying a number of diversity characteristics that affect an applications completion time. Initial experiments performed on a real world federated Cloud show that VMs running jobs deployed on homogenous OSs within heterogeneous Cloud infrastructure result in a difference in execution time comparable to that of VMs using diverse OSs within the same Cloud infrastructure, confirming the problem mentioned previously. These results reflect findings in related works, such as Cloud unpredictability [11] decreasing the effectiveness of real-time scheduling and platform heterogeneity [13] decreasing power efficiency in Cloud datacenters. This heterogeneity can lead to a design choice to sacrifice environmental diversity of fault-tolerant applications to meet deadlines.

Future work will involve exploring existing Real-Time fault-tolerant mechanisms and analysing how they perform when scaled in a federated Cloud environment. Additionally, we also plan to develop a predictive model of the execution time of a VM when deploying a specific OS using approximated provisioned resources. This will enable the design of an intelligent allocation mechanism that selects the most suitable VM configuration to be deployed in a fault-tolerant Real-Time Application to fulfill specified deadlines. Finally, we plan to investigate whether there is some sort of ideal balance between sufficient environment diversity for our RTA application to avoid unacceptable frequency and severity of service failure caused by failures and making execution times of heterogeneous VM deployments less unpredictable in the federated Cloud environment.

Acknowledgments

We would like to thank Beihang University for their assistance and permission in using the Beihang and Shahe Clouds.

References

[1] P. Mell, T. Grance. "The NISTdefinition of cloud computing". Technical report, National Institute of Standards and Technology, *2009*.

[2] Zibin Zheng; Zhou, T.C.; Lyu, M.R.; King, I.; , "FTCloud: A Component Ranking Framework for Fault-Tolerant Cloud Applications," *Software Reliability Engineering (ISSRE), 2010 IEEE 21st International Symposium on* , vol., no., pp.398-407, 1-4 Nov. 2010

[3] Rochwerger, B.; Breitgand, D.; Epstein, A.; Hadas, D.; Loy, I.; Nagin, K.; Tordsson, J.; Ragusa, C.; Villari, M.; Clayman, S.; Levy, E.; Maraschini, A.; Massonet, P.; Muñoz, H.; Tofetti, G.; , "Reservoir - When One Cloud Is Not Enough," *Computer* , vol.44, no.3, pp.44-51, March 2011

[4] R. Buyya, R. Ranjan, and R. Calheiros, Intercloud: Utility-oriented federation of cloud computing environments for scaling of application services. in ICA3PP 2010: The 10th International Conference on Algorithms and Architectures for Parallel Processing, 2010, pp. 19--24.

[5] S.Shankland "Amazon suffers U.S. outage on Friday" Internet. Available: http://news.cnet.com/8301-10784_3-9962010-7.html

[6] Shin, K.G.; Ramanathan, P.; , "Real-time computing: a new discipline of computer science and engineering," *Proceedings of the IEEE* , vol.82, no.1, pp.6-24, Jan 1994

[7] Ken Birman , Gregory Chockler , Robbert van Renesse, "Toward a Cloud Computing Research Agenda", ACM SIGACT News, v.40 n.2, June 2009

[8] Stankovic, J.A.; , "Misconceptions about real-time computing: a serious problem for next-generation systems," *Computer* , vol.21, no.10, pp.10-19, Oct 1988

[9] "Amazon EC2 Instance Types", Amazon, Internet. Available: http://aws.amazon.com/ec2/instance-types/

[10] "Cloud Sites Pricing and Scaling" Rackspace, Internet. Available: http://www.rackspace.com/cloud/

[11] Linh T.X. Phan, Zhuoyao Zhang, Qi Zheng, Boon Thau Loo, Insup Lee "An Empirical Analysis of Scheduling Techniques for Real-time Cloud-based Data Processing" *The 4th IEEE International Workshop on Real-Time Service-Oriented Architecture and Application (RTSOAA 2011)*, Irvine, USA, December 2011

[12] Vijay Sarathy, Purnendu Narayan, Rao Mikkilineni, "Next Generation Cloud Computing Architecture: Enabling Real-Time Dynamism

forShared Distributed Physical Infrastructure," wetice, pp.48-53, 2010 *19th IEEE International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises*, 2010

[13] Nathuji, R.; Isci, C.; Gorbatov, E.; , "Exploiting Platform Heterogeneity for Power Efficient Data Centers," *Autonomic Computing, 2007. ICAC '07. Fourth International Conference on* , vol., no., pp.5, 11-15 June 2007

[14] M. Birkin, P. Townend, A. Turner, B. Wu, J. Arshad, J. Xu, "MoSeS: A Grid-enabled spatial decision support system", in Social Science Computing Review, in press. DOI: 10.1177/089443930933229, 2009

[15] A. Celesti, *et al.* 'Three-phase Cross-cloud Federation Model: The Cloud SSO Authentication' *Published on Proceedings of The 2nd IEEE International Conference on Advances in Future Internet (AFIN 2010),* Venice, Italy July 2010.

[16] C. Cachin, R. Haas, and M. Vukoli´c. Dependable storage in the Intercloud. Research Report RZ 3783, IBM Research, Aug. 2010

[17] B. Randell et al., "Dependability – Its Attributes – Impairments and Means", in Predictably Dependable Computing Systems, Springer-Verlag, 1995

[18] Fadishei, H.; Saadatfar, H.; Deldari, H.; , "Job failure prediction in grid environment based on workload characteristics," *Computer Conference, 2009. CSICC 2009. 14th International CSI* , vol., no., pp.329-334, 20-21 Oct. 2009

[19] Y-M. Wang, Y. Huang, K.-P. Vo, P.-Y. Chung and C. Kintala, "Checkpointing and Its Applications," *Proc. 25th Int"l Symp. Fault-Tolerant Computing,* pp. 22-31,Pasadena, Calif., June 1995.

[20] I. Foster et al., "Cloud Computing and Grid Computing 360-Degree Compared," Grid Computing Environments Workshop (GCE '08), 2008.

[21] Paul E. Ammann , John C. Knight, Data Diversity: An Approach to Software Fault Tolerance, IEEE Transactions on Computers, v.37 n.4, p.418-425, April 1988.

[22] Avizienis, A.; Laprie, J.-C.; Randell, B.; Landwehr, C.; , "Basic concepts and taxonomy of dependable and secure computing," *Dependable and Secure Computing, IEEE Transactions on* , vol.1, no.1, pp. 11- 33, Jan.-March 2004