EVOLUTION ON THE LATENCY BENEFITS FOR MULTI-CLOUD VIRTUAL CLUSTER DEPLOYMENTS FOR MTC APPLICATIONS WITH EFFECTIVE RESOURCE UTILIZATION ON MTC APPLICATION

Ms. Leena Patel
Assistant Professor, Gandhinagar Institute of Technology
Gandhinagar, India.

Abstract:-* Cloud computing has gained greater importance in many IT organizations, as an elastic, flexible and variable-cost way to deploy their service platforms using outsourced resources. To minimize user-perceived latencies, web services are often deployed across multiple geographically distributed data centers. The premise of this work is that web services deployed across multiple cloud infrastructure services can serve users from more data centers than that possible when using a single cloud service, and hence, offer lower latencies to users. The cluster nodes can be provisioned with resources from different clouds to improve the cost-effectiveness of the deployment, or to implement high-availability strategies. Here I conduct a comprehensive measurement study to understand the potential latency benefits of deploying web services for Multi-Cloud Virtual Clusters for MTC applications. In this research paper it is explored with the scenario to deploy a computing Multi-cloud virtual cluster on top of a multi-cloud infrastructure, for solving loosely-coupled Many-Task Computing (MTC) applications which is yet in study only in recent research work up till. This paper defines the research on the latency benefits for multi-cloud virtual cluster deployments for MTC applications which is yet totally referred only in study and so it gives a brief direction to get it as a research for future work to be implemented.

***Keywords:* Cloud Services, Multi-Cloud Virtual Cluster, Latency, Loosely-coupled Multi Task Computing, Resource Utilization.**

I. INTRODUCTION

In a recent work, research is extended by including virtualization in the local site so providing a flexible and agile management of the whole infrastructure that may include resources from remote providers. However, all these cluster proposals are deployed using a single cloud, while multi-cloud cluster deployments are yet to be studied and so obviously the Multi-Cloud Virtual Cluster deployment is yet to be for evolution. Many-Task Computing (MTC) paradigm [1] embraces different types of high-performance applications involving any different tasks, and requiring large number of computational resources over short periods of time. These tasks can be of very different nature, with sizes from small to large, loosely coupled or tightly coupled, or compute-intensive or data-intensive. Cloud computing technologies can offer important benefits for IT organizations and data-centers running MTC applications:

Elasticity and rapid provisioning, enabling the organization to increase or decrease its infrastructure capacity within
minutes, according to the computing necessities; pay-as-you-go model, allowing organizations to purchase and pay for the exact amount of infrastructure they require at any specific time; reduced capital costs, since organizations can reduce or even eliminate their in-house infrastructures, resulting on a reduction in capital investment and personnel costs access to potentially “unlimited” resources, as most cloud providers allow to deploy hundreds or even thousands of server instances simultaneously and flexibility, because the user can deploy cloud instances with different hardware configurations, operating systems, and software packages. Computing clusters have been one of the most popular platforms for solving MTC problems, especially in the case of loosely coupled tasks (e.g. high-throughput computing applications) [1].
The frequent use of different cloud providers to deploy a computing cluster spanning different clouds can provide several benefits:

• *High-availability and fault tolerance*, the cluster worker nodes can be spread on different cloud sites, so in case of cloud downtime or failure, the cluster operation will not be disrupted. Furthermore, in this situation, we can dynamically deploy new cluster nodes in a different cloud to avoid the degradation of the cluster performance [1].

• *Infrastructure cost reduction*, since different cloud providers can follow different pricing strategies, and even variable pricing models (based on the level of demand of a particular resource type, daytime versus night-time, weekdays versus weekends, spot prices, and so forth), the different cluster nodes can change dynamically their locations, from one cloud Provider to another one, in order to reduce the overall infrastructure cost [1].

I. *Background and Setting*

Here three popular cloud services are considered in this paper: Amazon EC2, Microsoft Azure, and Google Compute Engine (GCE). I provide an overview of these cloud services and describe our envisioned deployment of web services across these cloud services EC2, Azure, and GCE operate on an Infrastructures-a-Service (IaaS) model. In each service, customers can rent virtual machines in different data centers, which are referred to as regions [2].

II. LATENCY BENEFITS OF MULTI-CLOUD DEPLOYMENTS

Latency benefits: To estimate latency benefits, we first compute the latency that every prefix would experience in seven scenarios for deploying a web service: only on EC2, only on Azure, only on GCE, combination of any two of the three cloud services, and across all three cloud services. In each scenario, for every 5 minute measurement round in our dataset, we estimate the latency for every prefix as outlined in the previous section. We then divide our measurement dataset into 5 partitions—one for every week—and in each week, It is computed for every prefix the median latency it experiences across all measurement rounds. Since all of our findings were largely identical across weeks, we present results from the middle of the 5 weeks. Figures 1(a), 1(b), and 1(c) show the estimated latency benefits that multi-cloud web service deployments will offer, as compared to deployments solely on Azure, EC2, or GCE; relative latency benefit is the latency benefit offered by a multi-cloud deployment as a fraction of the absolute latency seen with a single cloud deployment. We can see that multi-cloud deployments can offer significant latency gains, especially as compared to deployments solely on Azure or on EC2. Our latency estimates show that, as compared to single-cloud deployments, latencies to 20–50% of prefixes can be reduced by over 20% by having web services span all three cloud services. A 20% latency gain is significant because even the simple operation of loading a single web page requires several RTTs of communication between a client and the web server [2]. In addition, in all three single-cloud deployments, we see that combining two cloud services yields most of the latency gains that multi-cloud deployments can offer. Expanding deployments from two to three cloud services only marginally improves the latency benefits. In the case of Azure-only, EC2-only, and GCE-only deployments, expanding the deployment to GCE in the first two cases and to EC2 in the third case accounts for most of the increased geographical diversity and the routing inefficiency fixes that multi-cloud deployments have to offer [2].

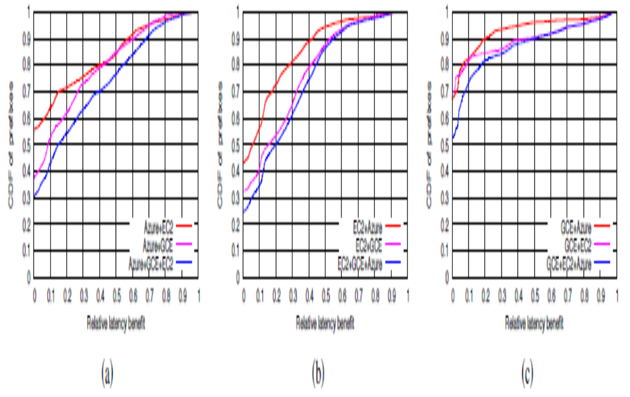


Fig.1: - Relative improvement in RTTs offered by multi-cloud deployments, as compared to deployments only on (a) Azure, (b) EC2, or (c) GCE [2].

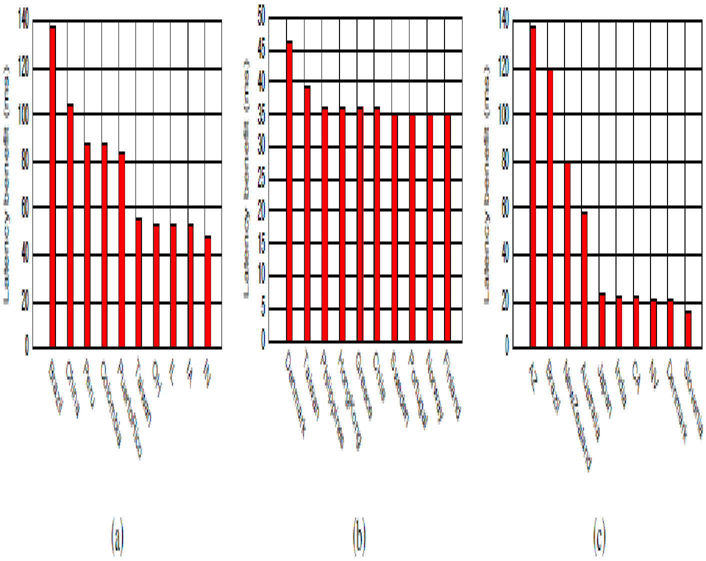


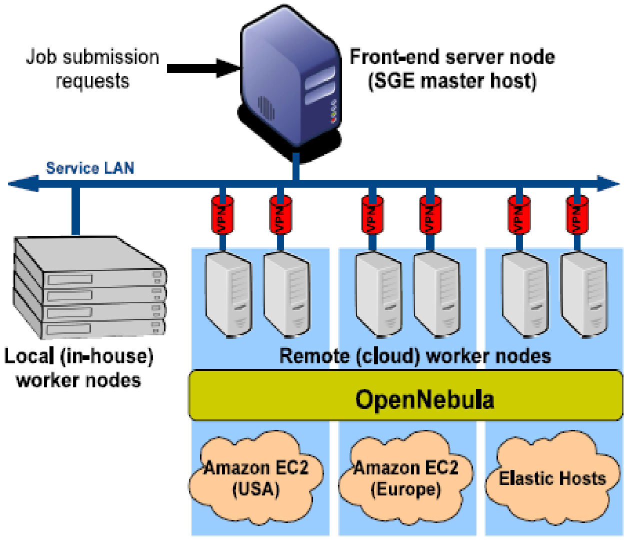
Fig.2. Median latency improvements expected in top 10 regions in which users benefit the most when web services are deployed across all cloud services as opposed to solely on (a) Azure, (b) EC2, or (c) GCE [2].

Biggest Gainers: - Figure 2 shows the top 10 regions that would experience the highest latency improvements when web services shift from single cloud deployments to deployments spanning EC2, Azure, and GCE. Here, It is computed that latency gain for a prefix as the difference in latency estimates for that prefix when using one cloud service and when using all three cloud services. In every region, the median of this latency gain is computed across all prefixes in that region. We can see that the biggest beneficiaries have latency improvements of over 100ms in the case of GCE and Azure, while several regions see more than 30ms reduction in latency in the case of EC2. As mentioned before, we consider these latency gains to be significant since the simple operation of loading a web page can involve tens of RTTs of interaction between a client and a server [4].

Reasons for latency gains:- As expected, the most common reason for reduced latency when using multiple cloud services is that one cloud service has a data center in a particular region while others do not. For example, users in Brazil would see latency improvements of over 100ms for web services currently deployed on GCE or Azure. This is because EC2 has a data center in Brazil, whereas GCE and Azure do not. Similarly, for web services currently deployed only on EC2, users in Hong Kong would have significant latency gains because Google has a data center in Hong Kong [2]. However, we also find that a significant fraction of latency gains are due to the ability of multi-cloud deployments to correct for routing inefficiencies. Due to inefficient routing, several regions have high latencies to a particular cloud service in spite of the presence of a nearby data center in that cloud service. Multi-cloud web service deployments can improve latencies in such cases because routing from the same region may be more efficient to nearby data centers in other cloud services [2].

III. DEPLOYMENT OF A MULTI-CLOUD VIRTUAL CLUSTER

Fig.3 shows the distributed cluster test bed used in this work deployed of top of a multi-cloud infrastructure. This kind of multi-cloud deployment involves several challenges, related to the lack of a cloud interface standard; the distribution and management of the service master images and the interconnection links between the service components [1].



Experimental test bed starts from a virtual cluster deployed in our local data center, with a queuing system managed by Sun Grid Engine (SGE) software, and consisting of a cluster front-end (SGE master) and a fixed number of virtual worker nodes (four nodes in this setup). This cluster can be scaled-out by deploying new virtual worker nodes on remote clouds. The cloud providers considered in this work are Amazon EC2 (Europe and USA zones) and Elastic Hosts. Table 1 shows the main characteristics of in-house nodes and cloud nodes used in the experimental test bed [1].

Table 1. Characteristics of different cluster nodes

Site	Arch.	Processor (single core) (USD/hour)	Mem. (GB)	Cost
Local data center (L)	i686 32-bits	Xeon 2.0GHz	1.0	0.04
Amazon EC2 Europe (AE)	i686 32-bits	Xeon 1.2GHz	1.7	0.11
Amazon EC2 USA (AU)	i686 32-bits	Xeon 1.2GHz	1.7	0.10
ElasticHosts (EH)	AMD 64-bits	Opteron 2.1GHz	1.0	0.12

IV. CONCLUDED PROPOSED SYSTEM FOR MULTI-CLOUD ENVIRONMENT TO CREATE VIRTUAL CLUSTERS WITH EFFECTIVE RESOURCR UTILIZATION ON MTC APPLICATIONS

EXISTING SYSTEM

In recently one server handles the multiple requests from the user. Here the server should method each of the request from the user at the same time, therefore the interval can be high. This could result in loss of information and packets could also be delayed and corrupted. On doing this the server cannot method the question from the user in an exceedingly correct manner. Therefore the interval gets magnified. It should results in traffic and congestion. For any application, software system should install within the consumer machine. Though sensible phones square measure expected to possess PC-like practicality, Hardware Resources like CPUs, Memory and Batteries square measure still restricted. Ancient utilities have solely Single supplier that is tougher to Support Multiple request [5].

In the Proposed System, It can be designed a Multi cloud Environment. Each Cloud Server will carry with Two Jobs. Cloud Server1 will process Job 1 & Job 2. Cloud Server2 will process Job 2 & Job 3. Cloud Server3 will process Job 2 & Job 3. If Client, requires for the Job 1 to the main Cloud Server. The Main Cloud Server will verify which Cloud Server is processing that Job 1 and it will also verify the load of both Cloud Server 1 and Cloud Server 3 as these servers will process Job 1. Based on the calculation of CPU Load for through put, the Main Cloud Server will determine the best Cloud Server for processing Job1. So Multi cloud Servers does the Jobs, it can be identifying the best Cloud Server for the data Process. For the Cloud Computing process, it is being implemented Cloud computing as Software as a Service (SAAS) and Infrastructure as a service (IAAS).For the SAAS, VLC Player is used for the service, for IAAS Data Query is used as a service deployment of Multi Cloud and is also coupled with Many-Task Computing (MTC). Multi cloud servers with Different Tasks are deployed to Identity the Best Cloud Server using its High Data Throughput [5].

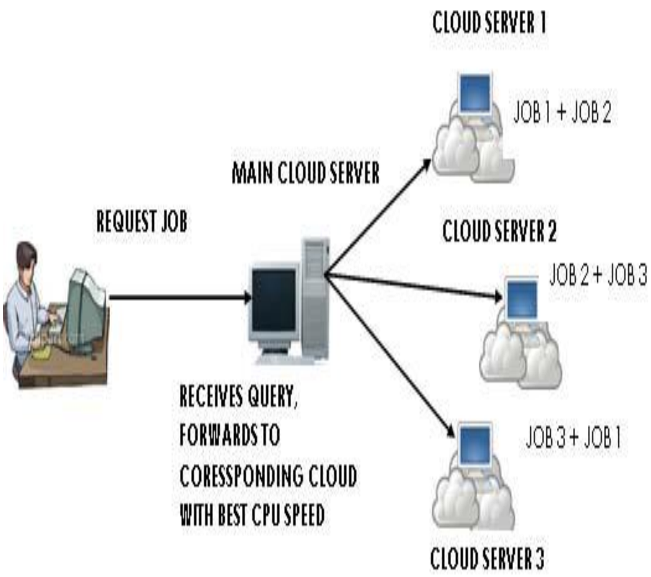
MODULE DESCRIPTION

A. Network Construction

This module is developed so as to make a dynamic network. In an exceedingly network, nodes area unit interconnected with the admin, that is observance all the opposite nodes. All nodes area unit sharing their data with every other.

B. Main cloud server

Client is system which sends the request to the main cloud server. Client details are verified & authenticated only then the client is allowed. The multiple clients can also send the request, but then the main cloud server will process the request one by one.



C. Cloud server (SAAS & IAAS)

The cloud server implementation during this project is software package as a Service (SAAS) and Infrastructure as a service (IAAS). The SAAS implementation is achieved victimization VLC Player. We tend to all perceive that while not VLC layer we tend to cannot play our computers. The software package as a Service (SAAS) is that the software area unit uploaded within the cloud server, once ever the consumer request the software package to the cloud server, the cloud server can give the software package. the most purpose of the IAAS is to fetch the file or knowledge requested by the user. Banking or hospital data area unit Store in cloud Server, consumer send the request to cloud server for any knowledge, and server are provided. This method are of use to cut back the consumer system load.

D. Resource Allocation to cloud server

We have to create a cloud server in the way, which will do multiple resources simultaneously and if multiple users is trying to access the cloud Server at a time, so the cloud server should be designed in the way that it should response simultaneously for all the users.

E. Client Request Processing Index File Maintenance

There will be multiple Cloud Servers which will have a main server and the main server will maintain Index will contain the data regarding the resources currently processing in the cloud servers. When a user requesting for a process to the cloud then the main server will verify the index file and then allocate the cloud to the user.

F. CPU Load Calculation

When a user requesting for a resource to the cloud server which was being processed by multiple cloud servers then the users resource will be allocate to the cloud server according to the CPU usage of the cloud server which was verified by the main server [5].

V. CHALLENGES IN HARNESSING LATENCY BENEFITS

Having focused thus far on the positives of deploying web services across multiple cloud services, I next highlight a couple of challenges in reducing user-perceived latencies in this manner.

Lack of control to improve poor latencies. Though we expect significant latency benefits for users in several regions, we can find that several other regions that experience high latencies with single-cloud web service deployments will continue to suffer from the same problem even when web services span EC2, Azure, and GCE. Figure 5 shows that the worst 20 regions (ranked based on the estimated latencies from EC2+Azure+GCE deployments) will all experience latencies over 50ms**1** in comparison, the latency for the median region is less than 25ms [2].

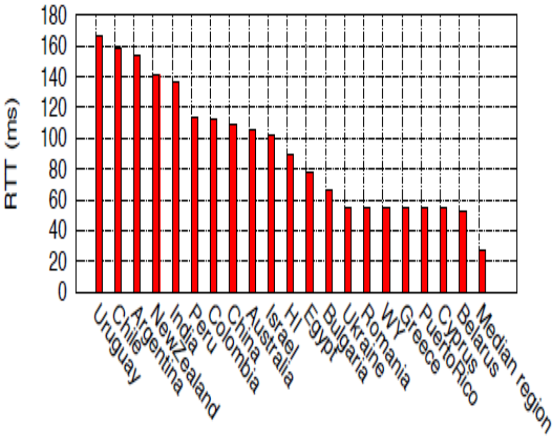


Fig.5: - The 20 worst regions with respect to the latencies from web service deployments spanning EC2, GCE, and Azure [2].

The most common reason for this lack of latency improvement is that the closest data center to a region is distant, even after combining cloud services. Chile, India, and Israel are examples of such regions. In other regions, in spite of the presence of a nearby data center in one of the cloud services, circuitous Internet routing to that data center causes high latencies to those regions even when combining cloud services. For example, though EC2 has a data center in Brazil, the latency to it is over 100ms from the Planet-Lab node in Argentina. Since neither Azure nor GCE has a nearby data center to correct this routing inefficiency, we see that users in Argentina will experience high latencies irrespective of whether web services use one or multiple cloud services [2].

Latency fluctuation. In the optimal redirection policy, in each measurement round in our dataset, a web service serves users in a prefix from that redirection option which has the lowest latency in that round to the prefix. We can find that these fluctuations in the lowest latency redirection option for a prefix are due to three main reasons. First, for many prefixes, latencies to the best and second best redirection options are largely identical and minor latency variations can alter the option that yields the lowest latency in any particular round; Figure 6(a) shows an example. On the other hand, there are several cloud service regions to which we see a distinct diurnal variation in latencies; this pattern is particularly dominant for Google's data centers in Europe as seen in Figure 6(b). In such cases, the lowest latency redirection option varies based on the time of day. Finally, I can also observe cases wherein we can see more long-term variations in latencies, e.g., Figure 6(c). We can believe that such cases are due to changes in routing configurations that either introduce or fix circuitous routing. Of the prefixes for which the best redirection option accounts for less than 80% of the measurement rounds in dataset, It is found that 32%, 46%, and 22% of the prefixes, respectively, can be attributed to the three above-mentioned reasons [2].

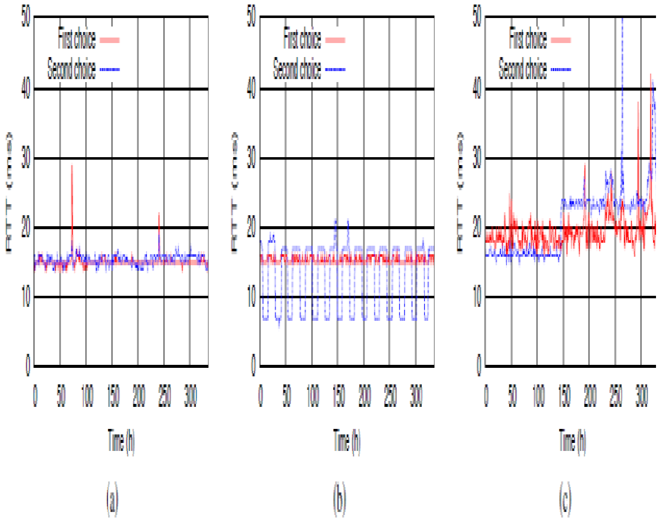


Fig.6. Examples of latency fluctuations that result in the lowest latency data center to a prefix changing over time [2].

It is observed here that the cloud service to which a prefix is redirected often changes over time has the following implication for the design of web services. Consider a user who may be served from data center A at some times and from another data center B at other times. The simplest design for a web service would be to store data uploaded by the user in one of these data centers, say A, and thus avoid the complications of replicating data.

However, as we see in Figure 6(b), if every user's data were stored only in the data center closest to the user, the 90th percentile latency experienced by that user will increase by 20% for users in 20% of prefixes. This is because, when the user is redirected to the alternate data center B, the web service incurs the overhead of fetching the user's data from A. Thus, to obtain the optimal latency benefits offered by multi-cloud deployments, it is critical that web services replicate a user's data asynchronously between the two data centers—typically in different cloud services—which offer the lowest latencies to the user [2].

VI. CONCLUSIONS

To the best of my knowledge, Perhaps I would be the first to recognize the opportunity for aggressively minimizing user-perceived latencies by deploying web services across multiple cloud services based on the proposed multi-cloud virtual cluster environment '*running for MTC application'*. In this paper I have also analyzed the challenges in harnessing latency benefits such that it can be useful to evolve Multi-Cloud Virtual Cluster System with MTC applications too moreover covering enough analysis on 'How can the effective resource utilization be possible on already proposed system for Multi-Cloud Environment running on MTC applications. To address these concerns, multi-cloud storage systems whereupon the data is replicated across multiple cloud storage services (potentially operated by distinct providers) have recently become a hot topic in the systems community [6,7, 8,9,10,11].This research paper will definitely give the direction to invent the multi-cloud virtual cluster environment deployments even with effective resource utilization working on MTC applications which is right now only in research study yet so far so it again may be the possible implementations for the future work for all IT professional and academicians in cloud computing area.

REFERENCES

[1] Rafael Moreno-Vozmediano, Ruben S. Montero, Ignacio M. Llorente, July 2010. Multi-Cloud Deployment of Computing Clusters for Loosely-Coupled MTC Applications. DRAFT FOR IEEE TPDS (SPECIAL ISSUE ON MANY-TASK COMPUTING), JULY 2010.

[2] Zhe Wu and Harsha V. Madhyastha, April 2013. Understanding the Latency Benefits of Multi-Cloud Web service Deployments. ACM SIGCOMM Computer Communication Review, Volume 43, Number 2, April 2013.

[3] IP2Location. http://www.ip2location.com/.

[4] M.Al-Fares, K.Elmeleegy, B.Reed, and I.Gashinsky. Overclocking the Yahoo! CDN for faster web page loads. In IMC, 2011.

[5] A.R. Arunachalam, G.Karthick, B. Rathinavel priyadasan, May, 2013. Implementation of Multicloud Computing Deployment System of SAAS and IAAS for Effective Resource Utilization on MTC Application. A.R. Arunachalam et al. / International Journal of Computer Science & Engineering Technology (IJCSET), ISSN: 2229-3345 Vol. 4 No. 05 May 2013.

[6] Gregory Chockler, Dan Dobre, Alexander Shraer. Implementing a Robust Multi-Cloud Store: Capabilities and Limitations, Position Paper Based on recent study.URL : http://workshop13.tclouds-project.eu/abstracts/implementing.pdf

[7] Hussam Abu-Libdeh, Lonnie Prince house, and Hakim Weather spoon. RACS: a case for cloud storage diversity. In Symposium on Cloud Computing (SoCC), pages 229–240, 2010.

[8] Cristina Basescu, Christian Cachin, Ittay Eyal, Robert Haas, Alessandro Sorniotti, Marko Vukolic, and Ido Zachevsky. Robust data sharing with key-value stores. In DSN, pages 1–12, 2012.

[9] Alysson Bessani, Miguel Correia, Bruno Quaresma, Fernando Andr´e, and Paulo Sousa. Depsky: Dependable and secure storage in a cloud-of-clouds. In European Conference on Computer Systems (EuroSys), 2011.

[10] TClouds Project. Privacy and resilience for Internet-scale critical infrastructures. http://www.tclouds-project.eu.

[11] Yunqi Ye, Liang Liang Xiao, I-Ling Yen, and Farokh Bastani. Secure, dependable, and high performance cloud storage. In Proceedings of the 29th Symposium on Reliable Distributed Systems (SRDS), pages 194–203, 2010.