Proceedings of the Fifth HPI Cloud Symposium “Operating the Cloud“ 2017

Andreas Polze, Max Plauth (Eds.)

Technische Berichte Nr. 122

des Hasso-Plattner-Instituts für
Digital Engineering an der Universität Potsdam

Technische Berichte des Hasso-Plattner-Instituts für
Digital Engineering an der Universität Potsdam

Technische Berichte des Hasso-Plattner-Instituts für
Digital Engineering an der Universität Potsdam | 122

Andreas Polze | Max Plauth (Eds.)

Proceedings of the Fifth HPI Cloud Symposium "Operating the Cloud" 2017

Universitätsverlag Potsdam

Bibliografische Information der Deutschen Nationalbibliothek
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über http://dnb.dnb.de/ abrufbar.

Universitätsverlag Potsdam 2018
http://verlag.ub.uni-potsdam.de/

Am Neuen Palais 10, 14469 Potsdam
Tel.: +49 (0)331 977 2533 / Fax: 2292
E-Mail: verlag@uni-potsdam.de

Die Schriftenreihe **Technische Berichte des Hasso-Plattner-Instituts für Digital Engineering an der Universität Potsdam** wird herausgegeben von den Professoren des Hasso-Plattner-Instituts für Digital Engineering an der Universität Potsdam.

ISSN (print) 1613-5652
ISSN (online) 2191-1665

Druck: docupoint GmbH Magdeburg

ISBN 978-3-86956-432-6

Zugleich online veröffentlicht auf dem Publikationsserver der Universität Potsdam:
URN urn:nbn:de:kobv:517-opus4-411330
http://nbn-resolving.de/urn:nbn:de:kobv:517-opus4-411330

Preface

Every year, the Hasso Plattner Institute (HPI) invites guests from industry and academia to a collaborative scientific workshop on the topic **Operating the Cloud**. Our goal is to provide a forum for the exchange of knowledge and experience between industry and academia. Co-located with the event is the HPI's Future SOC Lab day, which offers an additional attractive and conducive environment for scientific and industry related discussions. **Operating the Cloud** aims to be a platform for productive interactions of innovative ideas, visions, and upcoming technologies in the field of cloud operation and administration.

In these proceedings, the results of the fifth HPI cloud symposium **Operating the Cloud** 2017 are published. We thank the authors for exciting presentations and insights into their current work and research. Moreover, we look forward to more interesting submissions for the upcoming symposium in 2018.

Contents

Application Performance Monitoring on Distributed Web Applications

Keynote

Jan Graichen

Internet Technologies and Systems Group
Hasso Plattner Institute for Digital Engineering
University of Potsdam, Germany
jan.graichen@hpi.uni-potsdam.de

Web applications have become more and more complex, while larger deployments and distributed cloud applications provide new challenges for operators and developers. There is an ever-growing need for detailed insight in application and infrastructure behavior. The presented thesis compares and evaluates tools to collect, aggregate and visualize metrics, and outlines the design of a novel Application Performance Monitoring (APM) software. A prototypical implementation has been developed as open source software and is deployed to the openHPI online learning platform. By using the advanced features for the new APM software, such as a detailed behavior tracing, tracing of concurrent and parallel activity, and cross application tracing, several issues in the openHPI platform software have been identified and fixes been applied. Results show how metric monitoring and behavior tracing support operators and developers, and how openHPI benefits from a free, on-premise hosted solution.

Creating an Environment for Detecting Identity Deception

Estee van der Walt and Jan Eloff

Department of Computer Science
University of Pretoria, South Africa
estee.vanderwalt@gmail.com

In today's interconnected world we are all exposed to potentially harmful behaviour caused by fake identities, be they those of people or machines. It is difficult to discern whether you are communicating to another entity you can trust. Today fake identities and identity deception constitute a cybercrime threat to all people connected to cyber communities. Detecting Identity Deception in a small environment, say for example where there are 1000 people, is expected to be a feasible task. This is however not the case where the numbers of people in an environment are running into millions — if not billions — such as is the case on Social Media Platforms (SMPs). This paper focuses on answering the following question: "Is it at all possible to detect Identity Deception on big data platforms?" Furthermore, a discussion is provided about the type of environment that would be required to mine and process 'big data' in order to detect Identity Deception on social media platforms.

1 Introduction

According to IDC's annual Digital Universe study the volume of data in the digital world is set to grow to 44 zettabytes by 2020 [33]. Large volumes of data are described by the term 'big data': this term was first coined in the 1990s and made its first academic appearance in 1998 [10]. Big data also shows characteristics, known as the 3Vs, of high volume, velocity, and variety [21].

Social media platforms (SMPs), which are an example of big data, are however vulnerable to various cybercrime threats. Examples of such threats are: data interception [39], identity theft [19], identity deception [32] **[RN325]**, and online bullying [12] [28].

This paper focuses on Identity Deception as a cybercrime threat on SMPs. With Identity Deception, the presented or perceived identity of an entity is different from what is expected. The entity could be a person, or as in the case of SMPs, an online account. Identity Deception detection by a person has been well-documented in past research and psychologists are believed to be successful at detecting such deceptions [11]. Manual deception detection techniques have also proven fruitful in the field of criminology [36]. With SMPs this form of manual detection is however not feasible due to the steep volume of data, as well as to the complexity and heterogeneity of the data. Research therefore points to the use of automated Identity Deception techniques in the form of predictive modeling and / or machine learning [32] [9].

To successfully employ machine learning models for detecting deceptive identities a specialized technical environment is required. For example, one might consider the type of data required for detecting deceptive identities that include, amongst others, images of persons and their GPS locations. Past research has posited various environments for the processing of big data in general. Distributed computing environments like Hadoop [6] have been suggested, as have been environments for cloud computing scenarios [1]. Most of the environment choices have been divided into problems relating to either data complexity or computation complexity [18]. Because Identity Deception detection is perceived as both a complex data and a computation problem one wonders whether previous work like [6] and [1], that focus on creating big data environments, is usable for the study at hand. The data required for detecting deceptive identities is of heterogeneous nature and hence requires an appropriate infrastructure that can handle the types of data involved. In addition, many machine learning iterations are required during the training phase in order to find the most accurate model for predicting this form of deception. The environment should thus allow for the timely discovery of an accurate identification model over many iterations.

Section 2 of this article discusses an environment suitable for detecting Identity Deception. Hardware, software, and network requirements, amongst other components, are discussed. In this section the technical environment provided for the research at hand is presented. It is believed that this environment is generic enough to be employed for other research studies of a similar nature. Section 3 discusses results from the running of various "Identity Deception detection" experiments within this environment. This is aimed at revealing whether the available environment can perform Identity Deception detection. Section 4 discusses whether the research environment provided was sufficient to detect Identity Deception on SMPs and on how different environment choices affected the time required to reach an accurate prediction. The article concludes with a discussion on future research.

2 The Research Environment

To detect Identity Deception on SMPs, data is required to be mined and trained in order to detect such deception via the use of various machine learning models. Subsequently, the final Identity Deception mechanism, providing for example, an Identity Deception score, is to be designed, calculated, and visualized. To produce such a process providing for the development of a so-called Identity Deception score, infrastructure including platforms, hardware, operating and application software, SMPs and networking, is required [23].

Figure 3 demonstrates the infrastructure that was provided for the research at hand in order to detect Identity Deception on SMPs.

Figure 2 illustrates the flow of data throughout the infrastructure.

The infrastructure shown in figure 3 will be discussed in more detail (see below).

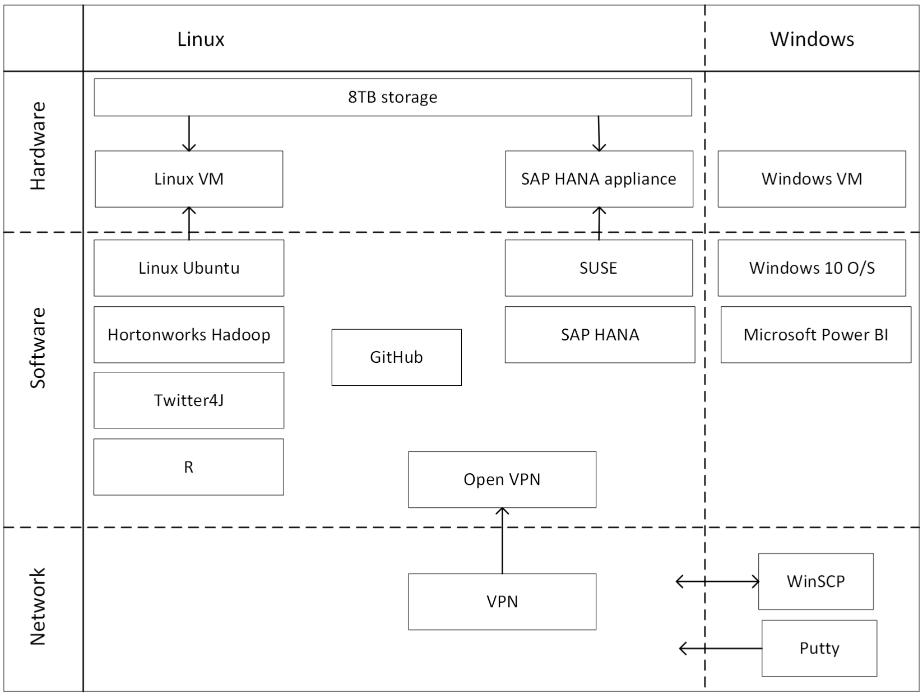

Figure 1: Research infrastructure

2.1 Hardware

Hardware from the HPI Future SOC (Service-Orientation Computing) lab was used [15]. This lab is part of the Hasso Plattner Institute (HPI) based in Potsdam, Germany. The lab provides researchers with access to hardware and software free of charge. All researchers are requested to submit a proposal for use of the infrastructure on a 6-monthly basis, together with an agreement to submit a research progress report after each expired period.

The HPI Future SOC research lab provided access to the following hardware, based on the proposal submitted towards the detection of Identity Deception on Social Media Platforms:

- Access to a SAP HANA server with 2TB RAM, 32CPUs / 100 cores. SAP HANA is a hardware appliance that loads data in memory for processing. This has great advantages in terms of returning results much faster than in the case of traditional relational databases reading data from disk first.

- A Linux virtual machine with 64 GB RAM, 4 CPUs / 8 cores. The RAM and CPUs are important for machine learning model training. Data are loaded in memory during training and the processing is split over the CPU cores. The more RAM and CPUs, the faster the training will be completed.

- 8TB storage space shared between the server and virtual machine. The storage is required to store the volumes of data that will be mined from SMPs.

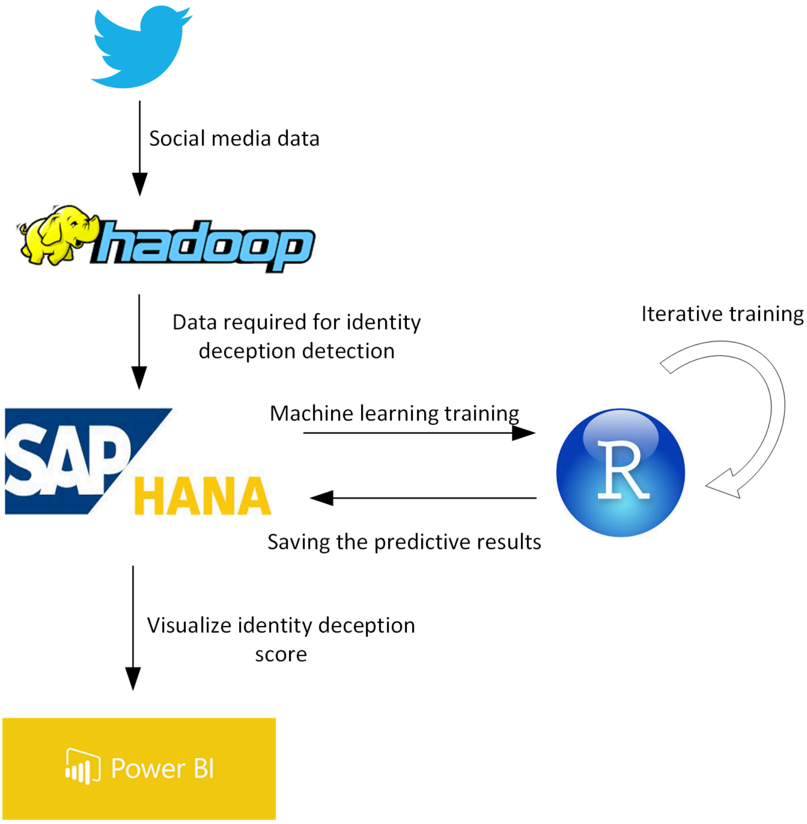
Social media data
hadoop
Data required for identity deception detection
Iterative training
Machine learning training
SAP HANA
R
Saving the predictive results
Visualize identity deception score
Power BI

Figure 2: Data flow to detect Identity Deception

In addition to the above, the final Identity Deception scores must be visualized. A standard Windows VM was also used for this purpose. The Windows VM had 16GB RAM and 1CPU / 4 cores.

2.2 Network

Communication with the HPI Future SOC lab domain was made possible via a Virtual Private Network (VPN). Researchers are provided with a unique username and password. The username is restricted to resources on the domain to which access has been granted. Once the VPN connection is established for the domain, hardware devices in the lab are accessible as if part of the local network.

2.3 Software

Various types of software components were used to access the network, use the hardware, and perform the research. The software used for this research performs various functions. Each of the software components used will be described next.

2.3.1 Social Media

To detect Identity Deception on SMPs, data was required. Twitter, being one of the top 6 SMPs [8], was chosen. The Twitter4j Java API [34] is available for free to any developer who aims to mine data from the social media platform. Help in using the API is readily available on the web [34]. The use of this API was preferred as both Hadoop and SAP HANA can interpret Java compiled code.

The meta data available in Twitter lends itself towards Identity Deception detection. Attributes like name, location, and profile image are mandatory to all account holders and describe the identity of a human [17]. Before this meta data could be used in the research, it was imperative to get ethical approval [40]. Ethical approval is vital when data is used in research containing information that could identify the human. For final ethical approval, the authors chose to inject artificially created human accounts in addition to the human information. The research focused on finding these artificially injected accounts.

2.3.2 Network Software

The following software was used to access the domain in Potsdam, Germany:

- For connection to the HPI Future SOC lab, the lab prefers the OpenVPN GUI software [38].
- For connecting, transferring of data, and configuration of the VM instance, PuTTY [29] and WinSCP [29] were used.

2.3.3 Operating System

A SAP HANA server and a Linux virtual machine were provisioned by the lab for the research at hand:

- The SAP HANA server had SUSE Linux installed as an operating system. This server was also shared with other research projects.
- The virtual machine was not shared amongst other researchers. Ubuntu Linux [7] was installed on the virtual machine. The reason for employment of this operating system is its compatibility with the SAP HANA server running SUSE Linux.

Lastly, a personal VM running on a Windows operating system was used in the research. The VM was used to visualize the final Identity Deception score.

2.3.4 Database

Two databases were used during the research, i.e. Hortonworks Hadoop and SAP HANA. The former [3] was primarily used to mine social media data and to handle the heterogeneous nature of the data. Hortonworks Hadoop [3] ran on the Ubuntu Linux [7] virtual machine hosted in the HPI Future SOC research lab in Potsdam, Germany. The following services were deployed with Hadoop in order to mine and query the data, and manage the cluster.

- Flume [30]: Flume is a streaming service. This service was configured in Hadoop to mine Twitter data using custom code written in Java.
- HDFS [27]:The mined Twitter data was stored in the Hadoop Distributed File System (HDFS).
- HBASE [27]: The mined Twitter data was also stored in HBASE to enable the performing of an initial exploratory data analysis.
- Hive [27]: Hive is the query tool that was used to interrogate the data.
- Ambari [4]: This was used for administration of the Hadoop cluster and for starting/stopping the services, like Flume.

The mined Twitter data was transferred to SAP HANA [26]. This could be done thanks to the in-memory high-performance capabilities of SAP HANA, which makes querying large data sets much faster than via Hadoop [5]. The XS Engine [25] from SAP HANA was used to inject streamed Tweets hosted in Hadoop and to populate the appropriate database tables. For connecting to the SAP HANA instance, SAP HANA Studio (Eclipse) [14] was used as development environment.

2.3.5 Machine Learning

The R language [16] was used to perform exploratory analysis. In addition, R also has many libraries for machine learning and data visualizations. In terms of machine learning, the Caret package [20] in R was used. For data visualization on the other hand, the ggplot2 library [37] was used to display the machine learning results in various graph representations. R Server [16] was installed on the Linux VM to enable R scripts to be run remotely.

2.3.6 Code Repository

All code was backed up to a GitHub [13] repository server in the cloud. This process was automated from the R development environment in order to ensure that code changes are always safely tracked and stored in the cloud.

2.3.7 Visualization Software

Microsoft PowerBI [22] was used in the research to visualize the final Identity Deception score. Microsoft PowerBI runs on the Windows operating system only. Owing to this fact, the scores were displayed on the Windows VM available to the research environment. Microsoft PowerBI was chosen due to its power in visualizing and quickly filtering data real-time.

3 Research Environment Considerations

For the research environment, speed was considered an important objective from the start. Many experiments had to be run during the research. The faster the execution, the more could be achieved. In earlier experiments [35] it took more than 3 days to train machine learning models. It was soon realized that if this time could be decreased, it would allow for more iterations of experiments. Speed is also important as the volume of data increases daily [31]. This calls for more dynamic detection mechanisms that can adapt to these data volumes [24]. Above all, it is imperative to build new Identity Deception detection models that can detect these new forms of deception in a timely manner.

Various experiments were run to determine how an environment that serves to detect Identity Deception on SMPs would allow for training machine learning models in a timely manner. The experiments considered the following:

- The operating environment – Linux and Windows were compared to understand whether one environment language could train machine learning models faster than another one.

- Dataset size – Two different dataset sizes were trained to understand effect on execution time. For the experiment both 50,000 and 150,000 records were tested.

- CPU cores used – One machine learning model's computations can be split across multiple CPUs. This experiment tested the effect on total execution time when 1 CPU was used as opposed to 6.

- Hyperparameter tuning – Different machine learning models such as random forest and support vector machines require different parameters as input. These parameters (also called hyperparameters [2]) can have various input values which affect the predicted model.

- Resampling repeat choice – The amount of times the machine learning model should repeat its training were experimented with. Every repetition of a machine learning model's training will use a different set of training data. By repeating the training, the best predictions can be determined. However, the repetitions influence the overall execution time.

- Resampling fold choice – To ensure that data inconsistencies are catered for, the data is divided into many folds. The machine learning model will use all but one fold to train the data and test the model's accuracy on the hold-out fold. The next iteration will hold-out a different fold. This process is repeated until all folds had the chance of being a hold-out fold. Figure 3 illustrates the iterations of a 3-fold process. The final model would be the average overall iterations.

The only persistent fact was that the data was hosted on the SAP HANA database. All these considerations were measured against time. Time was split into two components, i.e.:

- The time to query data on the SAP HANA database and have it available to train via a machine learning model.

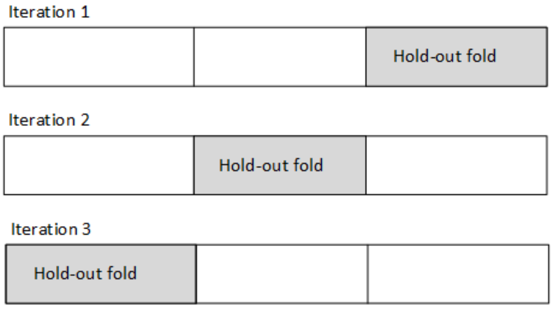

Figure 3: An illustration of 3-fold resampling

- The total time of all operations including the time to query the data, train a machine learning model, and return the results of the predictions.

Each consideration will be discussed next. In all cases the random forest machine learning model was executed in the R environment to test all conditions.

3.1 Operating System

Table 1 shows a comparison of running the machine learning algorithm under both the Windows and Linux operating systems.

Table 1: Operating system comparison

O/S	CPU	Size	Folds	Repeats	Hyper parameter	Query Time	Total Time
Linux	1	50K	5	0	3	6	1,983
Windows	1	50K	5	0	3	630	2,123

It is clear from the query time that the experiment with the Windows environment took a lot longer. This would make one think that the Linux environment is faster. This is however deceptive. The query time was significantly more for the Windows environment as the data had to travel from the Internet to reach the Windows machine, whereas the Linux VM is on the same Local Area Network (LAN) as the SAP HANA database. The Windows environment was faster in training machine learning models but due to the query time the overall Linux environment's performance was the best.

3.2 Dataset Size

Table 2 shows a comparison between running the machine learning algorithm for 50K and 155K+ records in a record set.

Table 2: Dataset size comparison

O/S	CPU	Size	Folds	Repeats	Hyper parameter	Query Time	Total Time
Linux	1	50K	5	0	3	5	5,120
Linux	1	155K+	5	0	3	21	5,157

It was expected that more data should take more time to query. It is interesting to note that the machine learning itself did not add much more time to the overall time.

3.3 CPU Cores

Table 3 shows a comparison of between running the machine learning algorithm using 1 core and using 6 cores of a CPU.

Table 3: CPU core comparison

O/S	CPU	Size	Folds	Repeats	Hyper parameter	Query Time	Total Time
Linux	1	155K+	5	0	3	29	17,603
Linux	6	155K+	5	0	3	23	6,791

As expected, more cores significantly reduced the overall machine learning time but did not add much value in querying the data.

3.4 Hyperparameters

Table 4 shows a comparison between running the same machine learning algorithm with the same dataset but a different amount of hyperparameters.

Table 4: Hyperparameter comparison

O/S	CPU	Size	Folds	Repeats	Hyper parameter	Query Time	Total Time
Linux	6	155K+	5	1	3	23	11,695
Linux	6	155K+	5	1	5	23	16,873
Linux	6	155K+	5	1	10	23	39,072

It was expected that the total machine learning time would increase when the number of hyperparameters to test are increased. The reason for this is that every additional hyperparameter to be tested, results in an additional machine learning run. This is also clear from the results. Trying 10 hyperparameters more than doubles the total machine learning time when compared to trying 5 hyperparameters.

3.5 Resampling Fold Size

Table 5 shows a comparison between running the same machine learning algorithm with the same dataset and a different number of folds.

Table 5: Resampling fold comparison

O/S	CPU	Size	Folds	Repeats	Hyper parameter	Query Time	Total Time
Linux	6	155K+	5	3	3	23	25,323
Linux	6	155K+	10	3	3	23	54,154

It was expected that increasing the folds would result in an increase in the total time. As more folds are introduced, more machine learning iterations are executed to train with the new folds. This is also clear form the results. 10 folds took more than double the total run time of 5 folds.

3.6 Resampling Repeat Count

Table 6 shows a comparison between running the same machine learning algorithm with the same dataset and a different number of repeats. It was expected that increasing the repeats would result in an increase in the total time. As more repeats are introduced, additional machine learning iterations are executed.

Table 6: Resampling repeat comparison

O/S	CPU	Size	Folds	Repeats	Hyper parameter	Query Time	Total Time
Linux	6	155K+	5	1	3	23	11,695
Linux	6	155K+	5	3	3	23	25,323
Linux	6	155K+	5	5	3	23	39,796

This is clear from the results. The total time increases linearly using 1, 3, and 5 repeats.

4 Findings with Regard to Creating an Environment for Detecting Identity Deception

Identity deception detection on SMPs was found to be possible on the environment provided. Hadoop provides the means to ingest the heterogenous volumes of data from SMPs. It became noticeable that the speed in building models for and detecting Identity Deception were influenced by the chosen environment. Although Hadoop was good at ingesting the data, the combination of R and SAP HANA allowed for a quicker detection of Identity Deception.

From the experiments run on a research environment to detect Identity Deception on SMPs, it was also clear that it is better for the data and machine learning engine to reside on the same LAN. This reduces overall query time.

Not only is the environment important but so too are the choices made for running experiments on the environment. It was found during this research that these choices cannot be ignored and that they should enable the identification of deception on SMPs. The most important choices comprise the number of available CPUs, repeats, hyperparameters, and resamples. The more CPUs, the quicker the machine learning training completes. The more repeats, hyperparameters, and resamples, the slower the machine learning training but more accurate the predictive model. There is of course a point where additional machine learning training will reach an optimal accurate predictive model and only add additional machine learning training time. A compromise should be found between time available for machine learning training and accuracy of the model.

5 Conclusion

The research environment was shown to be important for successful detection of Identity Deception on SMPs. The environment should be able to handle large volumes of heterogeneous data and consider the impact on machine learning model time. In addition, the ethical implications should not be forgotten when working with SMP data.

Although the focus of this paper was on creating an efficient experimental environment for detecting Identity Deception, additional insights were gained into the behaviour of deceptive humans. It was found that humans deceive by changing their username, profile image, or their location amongst others. The research environment allowed for many iterations to be run, testing various meta data combinations.

Future research will investigate which meta data attributes contribute towards Identity Deception on SMPs and how this contribution can be quantified.

Acknowledgements

The SAP HANA instance, virtual machine, and storage were provided by the HPI Future SOC research lab. The HPI Future SOC research lab facilities are ideal for researching the detection of Identity Deception on SMPs. We would like to thank the HPI Future SOC research lab for their assistance and support.

References

[1] D. Agrawal et al. *Challenges and Opportunities with Big Data*. White Paper. 2012.

[2] D. Anguita, L. Ghelardoni, A. Ghio, L. Oneto, and S. Ridella. "The 'K'in K-fold Cross Validation". In: *Proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium)*, pages 441–446.

[3] Apache Software Foundation. *The Hadoop Distributed File System: Architecture and Design*. 2014.

[4] S. S. Aravinth, A. Haseenah Begam, S. Shanmugapriyaa, S. Sowmya, and E. Arun. "An Efficient HADOOP Frameworks SQOOP and Ambari for Big Data Processing". In: *IJIRST — International Journal for Innovative Research in Science & Technology* 1.10 (2015), pages 252–255. ISSN: 2349-6010.

[5] M. D. Assunção, R. N. Calheiros, S. Bianchi, M. A. Netto, and R. Buyya. "Big Data Computing and Clouds: Trends and Future Directions". In: *Journal of Parallel and Distributed Computing* 79 (2015), pages 3–15. ISSN: 0743-7315.

[6] G. Bello-Orgaz, J. J. Jung, and D. Camacho. "Social Big Data: Recent Achievements and New Challenges". In: *Information Fusion* 28 (2016), pages 45–59. ISSN: 1566-2535.

[7] Canonical Ltd. *Ubuntu is an Open Source Operating System Software for Computers*. URL: `https://www.ubuntu.com/`.

[8] D. Chaffey. "Global Social Media Research Summary 2016". In: *Smart Insights: Social Media Marketing* (2016).

[9] S. Cresci, R. Di Pietro, M. Petrocchi, A. Spognardi, and M. Tesconi. "Fame for Sale: Efficient Detection of Fake Twitter Followers". In: *Decision Support Systems* 80 (2015), pages 56–71. ISSN: 0167-9236.

[10] F. X. Diebold. *A Personal Perspective on the Origin(s) and Development of "Big Data": The Phenomenon, the Term, and the Discipline*. 2012.

[11] P. Ekman, M. O'Sullivan, and M. G. Frank. "A Few Can Catch a Liar". In: *Psychological science* 10.3 (1999), pages 263–266. ISSN: 0956-7976.

[12] M. A. Al-garadi, K. D. Varathan, and S. D. Ravana. "Cybercrime Detection in Online Communications: The Experimental Case of Cyberbullying Detection in the Twitter Network". In: *Computers in Human Behavior* 63 (2016), pages 433–443. ISSN: 0747-5632.

[13] GitHub Inc. *GitHub Website*. URL: `https://github.com/`.

[14] M. A. Haque and T. Rahman. *Sentiment Analysis by Using Fuzzy Logic*. 2014.

[15] Hasso Plattner Institute for Digital Engineering. *The HPI Future SOC lab*. URL: `https://hpi.de/en/research/future-soc-lab.html`.

[16] R. Ihaka and R. Gentleman. *R: The R Project for Statistical Computing*. 2017.

[17] R. Jamieson, D. Winchester, G. Stephens, and S. Smith. "Developing a Conceptual Framework for Identity Fraud Profiling". In: *European Conference on Information Systems (ECIS)*. Citeseer, 2008, pages 1418–1429.

[18] X. Jin, B. W. Wah, X. Cheng, and Y. Wang. "Significance and challenges of big data research". In: *Big Data Research* (2015). ISSN: 2214-5796.

[19] M. Kabay, E. Salveggio, R. Guess, and R. D. Rosco. "Anonymity and Identity in Cyberspace". In: *Computer Security Handbook, Sixth Edition* (2014), pages 70.1–70.37. ISSN: 1118820657.

[20] M. Kuhn. "A Short Introduction to the Caret Package". In: *R Found Stat Comput* (2015), pages 1–10.

[21] D. Laney. *3D Data Management: Controlling Data Volume, Velocity and Variety*. URL: `http://blogs.gartner.com/doug-laney/files/2012/01/ad949-3D-Data-Management-Controlling-Data-Volume-Velocity-and-Variety.pdf`.

[22] Microsoft. *Microsoft Power BI*. URL: `https://powerbi.microsoft.com/en-us/`.

[23] V. Onomza Waziri, J. K. Alhassan, I. Ismaila, and M. N. Dogonyaro. "Big Data Analytics and Data Security in the Cloud via Fully Homomorphic Encryption". In: *International Journal of Computer, Control, Quantum and Information Engineering* 9.3 (2015).

[24] A. Ortigosa, J. M. Martín, and R. M. Carro. "Sentiment analysis in Facebook and its application to e-learning". In: *Computers in Human Behavior* 31 (2014), pages 527–541. ISSN: 0747-5632.

[25] SAP SE. *SAP HANA*. URL: `https://www.sap.com/developer/topics/sap-hana.html`.

[26] SAP SE. "SAP HANA predictive analytics library". In: *SAP HANA Platform SPS 10* Document Revision 1.0 (2015).

[27] K. Shvachko, H. Kuang, S. Radia, and R. Chansler. "The Hadoop Distributed File System". In: *Mass Storage Systems and Technologies (MSST), 2010 IEEE 26th Symposium on*. IEEE, pages 1–10.

[28] D. Smit. “Cyberbullying in South African and American schools: A legal comparative study”. In: *South African Journal of Education* 35.2 (2015), pages 01–11. ISSN: 0256-0100.

[29] S. Tatham. *PuTTY*. URL: `http://www.putty.org/`.

[30] S. G. Teo, S. Han, and V. C. S. Lee. “Privacy Preserving Support Vector Machine Using Non-linear Kernels on Hadoop Mahout”. In: *16th International Conference on Computational Science and Engineering*. 2013, pages 941–948.

[31] M. Tsikerdekis and S. Zeadally. “Detecting and Preventing Online Identity Deception in Social Networking Services”. In: *Internet Computing, IEEE* 19.3 (2015), pages 41–49. ISSN: 1089-7801.

[32] M. Tsikerdekis and S. Zeadally. “Multiple Account Identity Deception Detection in Social Media Using Nonverbal Behavior”. In: *Information Forensics and Security, IEEE Transactions on* 9.8 (2014), pages 1311–1321. ISSN: 1556-6013.

[33] V. Turner, J. F. Gantz, D. Reinsel, and S. Minton. “The digital universe of opportunities: Rich data and the increasing value of the internet of things”. In: *IDC Analyze the Future* (2014).

[34] Twitter Inc. *Twitter API*. URL: `https://dev.twitter.com/overview/api`.

[35] E. Van der Walt and J. Eloff. “Protecting minors on social media platforms - A Big Data Science experiment”. In: *HPI Cloud Symposium "Operating the Cloud"* (2015).

[36] G. A. Wang, H. Chen, J. J. Xu, and H. Atabakhsh. “Automatically detecting criminal identity deception: an adaptive detection algorithm”. In: *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on* 36.5 (2006), pages 988–999. ISSN: 1083-4427.

[37] H. Wickham. *ggplot2*. URL: `http://ggplot2.org/`.

[38] J. Yonan. *OpenVPN*. URL: `https://openvpn.net/`.

[39] Z. Zhang and B. B. Gupta. “Social media security and trustworthiness: overview and new direction”. In: *Future Generation Computer Systems* (2016). ISSN: 0167-739X.

[40] M. Zimmer and N. J. Proferes. “A topology of Twitter research: Disciplines, methods, and ethics”. In: *Aslib Journal of Information Management* 66.3 (2014), pages 250–261. ISSN: 2050-3806.

Cloud Identity Management – Current Issues and Developments

Isaac Odun-Ayo, Charles Ayo, and Nicholas Omoregbe

Department of Computer and Information Sciences
Covenant University, Nigeria
firstname.lastname@covenantuniversity.edu.ng

Cloud computing is a dynamic paradigm based on sound technology aimed at supporting IT users in essential computing task. Cloud providers offer services that can be used to carry out common task that is accessible online anywhere. Nevertheless, they come with some major challenges. The most common is that of security, which has made identity management necessary thereby compelling the Cloud service providers to ensure that users are properly authenticated. This paper presents the state of the art from some literature available on cloud identity management. The study was executed by means of review of some literature available on cloud identity management to examine the present trends towards providing a guide for future research in cloud computing.

1 Introduction

"Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction" [13]. Cloud computing is set to become the main focus of all activities around IT utilization. Cloud computing has the primary component to assist an enterprise and even SMBs to succeed in their IT endeavours. Cloud computing provide scalable, elastic, on demand services via the Internet to cloud users. The cloud utilizes the concepts of virtualization and multi–tenancy to deliver metered services to consumers through cloud service providers (CSPs). Cloud services are provided at three levels of abstraction, namely Software-as-a-Service (SaaS), Platform-as-a-Service (PaaS) and Infrastructure-as-a-Service (IaaS). In SaaS, services and applications are delivered over the Internet and accessed using a web browser by the users. The cloud service provider ensures the deployment and maintenance of the applications, the operating system and other cloud services [17]. In PaaS, the service provider offers a platform for users to create and deploy applications. The implication of this is that the user has little or no control over the operating system and other cloud resources, but manages and runs an application on the virtual resources provided by the service provider [17]. IaaS provides the user with the cloud environment to utilize processing, storage and bandwidth, making it possible for the user to run operating systems on virtual machines of the cloud provider [10].

Cloud computing has four deployment models that describe the scope of service offered to the cloud customers. They are the private, public, community and hybrid clouds. Private clouds are usually owned and hosted by an organization. The infrastructure may be on premise or off premise, but it is secure because users belong to that organization [17]. Public clouds involve cloud providers that offer services to customers over the Internet. They own and control massive data centers and other infrastructure, sometimes spread over different geographical locations. Security and privacy is a concern on public clouds. Community cloud is owned by several organization with shared common interest. The infrastructure is shared among the organization based on agreed policies. Hybrid clouds are a combination of either private, public, or community clouds. The entities are unique, but managed by a single unit [17].

Digital identity allows an entity to be represented in some forms of information that makes the entity recognizable within a particular framework [10]. Identity management (IDM) is a collection of related policies that allows the administration, maintenance, management, information exchange, discovery and authentication process used to identify an information with a view to ensuring overall security [10]. Information can be accessed at any place and any time over the Internet using cloud services. Therefore, there is the need to have an identity management scheme to verify a valid user and offer services based on such valid authentication credentials. An identity management process aims to secure the user and other processes in terms of private and sensitive information. Every enterprises is expected to restrict access to computing resources and sensitive information [6]. Cloud identity management systems are distinct from traditional methods because of unique characteristics of the cloud such as access control, scalability, virtualization, provisioning and multi-tenancy [9].

Identity management systems (IDMS) allow some important functions such as access rights, authentication and authorization to be given the deserved attention on the cloud [9]. A cloud IDM must secure virtual devices, dynamic machines and control points among others. Cloud services and deployments are scalable and elastic making it possible to launch servers, start services, terminate the servers and services, and also manage the assignment and reassignment of IP addresses in a dynamic manner [7]. Certainly, traditional IDMS that simply manage users and services in a fairly static manner cannot be suitable for cloud purposes. For example, it is critical to inform the IDM when a service has been terminated or a machine de-provisioned so that it can revoke future access [7]. IDM is expected to store details of such inactive processes until they become active again. Therefore, access management to the data of such devices becomes critical and must be in line with the service level agreement [7]. Again, normal IDM cannot be utilized on the cloud because of the unique nature of cloud operations and infrastructure. The purpose of this paper is to examine cloud computing and identity management systems. The paper discusses issues in IDM and thereafter highlights current IDM trends in industry. The remaining part of the paper is organized as follows: Section 2 discusses related work. Section 3 examines types of IDM and IDM architectures. Section 4 highlights industry IDM trends. Section 5 concludes the paper and makes suggestions for future work.

2 Related Work

Security in cloud computing: opportunities and challenges in [1] proposed a cloud computing architectural framework. Security challenges at various abstractions of cloud computing were examined. Identity management and access control were also examined in some details. Cloud computing security issues and challenges: a survey in [17] presented a survey of security issues in terms of cloud delivery and deployment modes. It also examined identity management in the area of cloud computing. Multi-tenancy authorization system with federated identity for cloud-based environments using Shibboleth in [11] proposed a framework for identity management using Shibboleth. The main focus was to provide identity management to enhance authentication and authorization in cloud computing. Assessment criteria for cloud identity management systems in [9] proposed a criteria for assessing cloud identity management systems. An identity management system is essential for access control, hence a comparative analysis will further enhance security on the cloud. Integrated Federated Identity Management for Cloud Computing in [16] discussed the identity management based on the different cloud layers. An integrated federated identity management was proposed and implemented. The benefits of the model were outlined in the paper. ICEMAN: an architecture for secure federated inter-cloud identity management in [4] considered the fact that no identity management schemes exist at the inter-cloud level. It proposed an inter-cloud IDM cloud security. Consolidated Identity Management System for secure mobile cloud computing in [10] observed that mobile users store personal information in an insecure manner on mobile devices.

Current IDMS for mobile devices have limitations making them vulnerable. A consolidated IDM is proposed to mitigate the vulnerabilities. Privacy-preserving digital identity management for cloud computing in [2] used encryption techniques for digital identity management. The proposed model allows verification of a user identity on multiple clouds. Identity and access management as security-as-a-service from clouds in [14] are a great enhancement of the identity management system. The identity and access management are distinct and are implemented as a service on the cloud. Identity in the cloud in [15] discussed the simple cloud identity management, which is a recent standard being adopted by cloud service providers. This model was developed by the open web foundation. A novel virtual identity implementation for anonymous communication in cloud environments in [6] seeks to further secure the process of identity management on the cloud. The approach is to hide the user′s identity by authenticating an unknown user in an unknown environment. Centralized end-to-end identity and access management and ERP systems: A multi-case analysis using the Technology Organization Environment framework in [3] proposed a model that enhances authentication and security both for an organization and the employees. The model is implemented and results have significant implication on the identity management of users.

3 Functions and Classifications of Identity Management Systems

3.1 Layers of Identity Management Systems

IDMS comprise the identity of the user used throughout a certain period based on software and components used to address such a user. It involves three main types of entities, namely the user, identity provider (IdP) and the service provider (SP) [10]. The following are the main functions of IDMS [10] [2]:

- **Account Management.** The process of providing identity within an enterprise is used for the users, administrators and developers in terms of provisioning and de-provisioning. This form of account management enables the provisioning of users and groups in different systems.
- **Authentications.** It is essential to determine who a person claims he / she is through the process of authentication. The process allows the user to be identified using various methods such as biometrics, login and user passwords or by means of some secret credentials.
- **Authorization.**This is meant to enable various kinds of level of access for different parts or operations within a computing system. It is the process of determining the permissions or rights that users have.
- **Federation.** This a process where organizations agree to collectively share authentication information based on trust. The identity of users is shared beyond the confines of an individual organization. Federation enables an authentication process on a global scale.
- **Auditing.** This is simply an accounting process that allows activities to be monitored so that occurrences can be traced to particular users when the need arises.

3.2 Classification of Identity Management Systems

Cloud IDMS are used to represent and recognize the identities in the digital world. IDMS can be classified into 4 distinct categories. Namely isolated, centralized, federated and anonymous. They are discussed in the subsequent paragraphs [9].

3.2.1 Isolated Identity Management System

Isolated IDMS makes use of a single server for the service provider and also the identity provider and it manages the identity information in storage for the user including user activities. This method is usually adopted by small and medium businesses [9]. An isolated IDMS is shown in Figure 1. Before getting service, the user is expected to complete the identity process at CSP1 and thereafter, CSP1 directs the user's request to its IDP for further attention. After a successful authentication process, a response is provided to the user. The IDM does not need any third party assistance for credentials and verification in this method. On the other hand, the process becomes cumbersome with increased services because each service must ascertain the credentials of users that have been authorized [9].

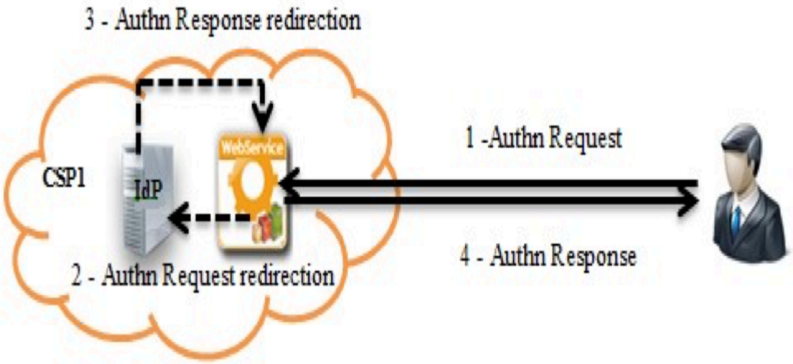

Figure 1: Isolated Identity Managing System [9]

3.2.2 Centralized Identity Management System

Centralized IDMS separate the functions of service providers and identity providers. In this case, management and storage of identity data including resources is handled by a third party single IdP that is trusted. The process starts with the collection of identity data by the IdP from the CSP in respect of the users. Thereafter, a cloud user may send an authentication request and it is routed to the relevant IdP who provide the necessary authentication response as shown in Figure 2.

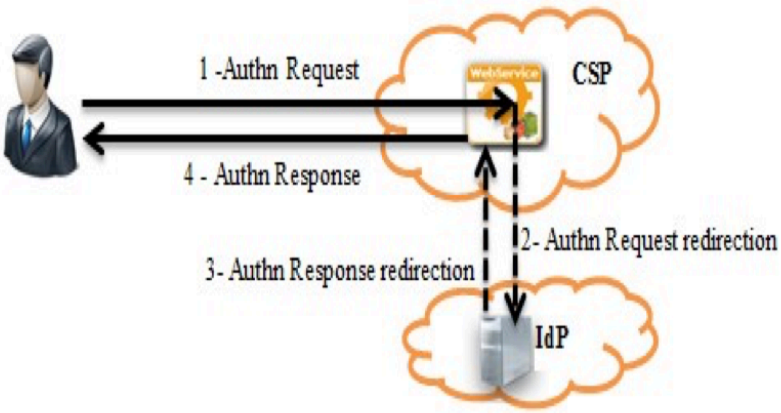

Figure 2: Centralized Identity Management System [9]

3.2.3 Federated Identity Management System

Federated IDMS allow subscribers from several organizations to utilize the same identity data for obtaining access to the networks that exist in group of trusted enterprises. The federated management system is enjoying wide usage because it eliminates the need for individual authentication accounts. The systems enable access across several domains to user credentials by external parties. The Federated IDM makes it possible to store identity information in different locations using the distributed storage architecture. As shown in Figure 3, a user may wish to carry out the authentication process through CSP1 before getting service. At CSP1, the process commences using the push and pull method that enables the CSP to access identity credentials stored by multiple service providers and IdPs. Although a service provider can maintain its own individual database to manage user credentials, the CSP must however link that information through the user's identity during the process of authorization and authentication.

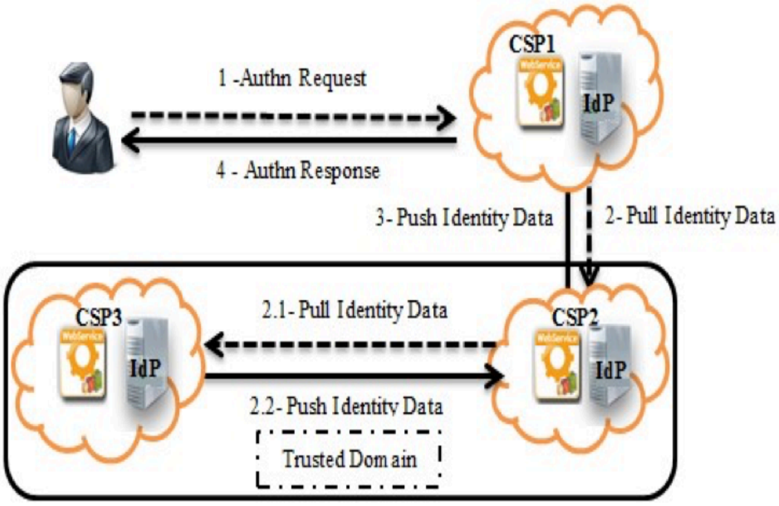

Figure 3: Federated Identity Managing System [9]

3.2.4 Anonymous Identity Management System

In anonymous IDMS, the user credentials can be kept relatively secret from users and providers alike. This is because there is no name associated with the identity information making it anonymous. It is however essential that the anonymity should be sufficiently strong to prevent leakage. Such leakage may not be deliberate, but could be a result of credentials being linked with other information that may be published.

3.3 Shibboleth

The OASIS security assertion makeup Language (SAML) introduced a system that allows for exchange of security information among organizations online. The SAML assertions is a portable one that enables trusted utilization among applications in different domains for the purpose of security. The OASIS SAML framework provides rules for using these SAML assertions to request, create and communicate [10]. Shibboleth [10] uses the process of federation identity based on SAML for its authentication and authorization processes. The user has the privilege of having a single sign on that is available across multiple domains. These organizations belong to the same federation and share identity credentials. The Shibboleth has the IdP and SP systems. The IdP manages the user authentication and also maintains the user credentials. In addition, it is responsible for proving user credentials to trusted organizations within the federation. Shibboleth has four components:

- **Handle Service.** This service is responsible for the authentication mechanism. In addition, it creates the token which is the SAML assertion that takes credentials to the users and also enables an organization to select appropriate authentication mechanism.
- **Attribute Authority.** This component handles the service providers request concerns. It handles attributes with applications of relevant privacy policies. It also enables users determine who can access them and proving a choice of directory to organizations.
- **Directory Service.** This is external to Shibboleth and it stores user attributes locally.
- **Authentication Mechanism.** This mechanism is not within Shibboleth and allows user authentication with the central service using a login / password. SP Shibboleth stores the resources required by the user for access. It also manages access control based on request by the IdP. An SP may comprise various applications, but it will be considered a single entity by the IdP. The SP Shibboleth has three main services:
  - **Assertion Customer Service.** This is responsible for receiving messages to (SAML) to establish a secure environment.
  - **Attribute Request.** It is responsible for obtaining and passing user attributes for the resources manager.
  - **Resources Manager.** It intercepts requests for resource and makes decisions to central access based on user attributes.

3.4 Identity Managing System Vulnerabilities

Some vulnerabilities have been identified in the layers of IDMS [16]:

- The first vulnerability lies in the compromising of IDM servers. By compromising an IDM server, an attacker can capture any token from within the IDM servers. IDM server compromise is a serious vulnerability that could create middle scale attacks against all users of the compromised IdP. This threat can be realized through malicious insider involvement or malicious code injection.

- The second vulnerability lies in the ease of mobile device capture or compromise through theft, loss or malicious mobile code injection. The impact is that private data on a mobile phone is exposed because most people store password and other sensitive information on mobile phones. Any data or services that can be accessed through the credentials on the phone become vulnerable.

- There is also a third possibility of intercepting and crypto-analyzing IDM message while being exchanged during the process of trust establishment between the user and CSP. Such unauthorized access can be used to gain entry to cloud services and data.

4 Identity Management System Industry Perspective

4.1 Identity Access Management-as-a-Service

Identity Access Management-as-a-Service (IDaaS) is a process to ensure secure IDM. IDaaS is useful in the area of machine learning, management and other security data. Most identity challenges lie with SaaS apps, hence IDaaS has been focused on its use for enabling cloud-based SaaS applications. Attention is now moving to using IDaaS to secure on premise applications. IDaaS will be useful on premise for synchronizing user directories, syncing password hashes and authentication users to active directory federation services. Using IDaaS to control access to on premise applications has several benefits. It will allow for the consolidation of visibility, control and policies in one place [12]. By 2019, 40% of IDaaS implementation will replace on-premise IDM implementations [8]. This increases the use of IDaaS in part stems from the difficulty and expense of running on premise IDM infrastructure. In addition, the growing use of other something-as-a-service offerings will make the decision widely accepted. Also, web and mobile applications will create a natural opportunity for the transition from in-house IDM to IDaaS. It was also predicted in the eGuide that by 2019, use of password and tokens will drop by 55% due to the introduction of recognition technologies. With the cost and accuracy of biometrics, they are offering good opportunities for use continuously in authentication.

4.2 Involvement of Cyber Security Experts in Identity Management System

IDM involves a lot of tools such as access policy, multiple data responsibility, manual process and others. User authentication continues to be anchored on user names and password, making nearly every organization vulnerable to credential harvesting, identity theft and cyber-attacks [8]. Enterprise Strategy Group recently undertook a research project and published the result in a report titled: "A Cyber Security Perspective on Identity and Access Management". Cyber security experts have become more involved with IDM for the following reasons:

- 36% say that involvement is the best practice in order to improve risk management and security best practices.
- 36% say it is to better detect things like credential theft, remote access and illegitimate account provisioning often associated with cyber-attacks.
- 33% say it is to improve regulatory compliance.
- 33% say it is because their organization had opened more internal applications and services to external users.
- 31% say because of the increasing use of cloud and mobile computing in their organization.

Organizations also foresee certain development with respect to IDM. Organization will consider more enterprise IDM project. IDM is fraught with security vulnerabilities and operational overhead. Security requirement may serve as a catalyst for strategic multimillion dollar IDM projects not seen since the 1990s. Enterprises will be more open to cloud-based control plans. SaaS offering for IDM such as Centrify, Okta, Ping etc. may become more attractive in an attempt to seek solutions to unify tools in IDM for cloud-based activities. Username / password authentication will be one of the first things to go. Everyone knows that username / password is a security nightmare, but it has proven too expensive to replace this method with security token or smart cards in the past. Wide support for fido specification and pervasive biometrics built into mobile devices may finally terminate username / password authentication. IDM skills will become more valuable and rare. Enterprise organizations will have more IDM architects and engineers, but may not find enough because the skills set are rare. IDM service specialist like Accenture, E&Y, HP, PWC and Unisys will bridge the gap and benefit financially in the process.

4.3 Enhancing Identity and Access Management

Security-wise the cloud is inherently risky, and many traditional identity and access management systems do not translate well to it [5]. Most organizations use out-of-date password policies, for example eight characters with complexity that do little to defend against the methods that hackers use to gain access to passwords. The correct way to defend against password guessing is to passphrase. However, even those organizations that correctly implement passphrase may find it difficult to extend this policy to the cloud since many cloud applications enforce a maximum password length.

Password guessing is not the only method hackers will use to gain access to accounts, credential theft is usually simpler via email phishing, a tactic favored by nation state and organized crime attackers. Therefore, cloud IDM policies need to reflect this risk and organizations should do their best to mitigate this by enforcing multi-factor authentication on all Internet facing services. To enhance cloud IDM, the organization must ensure that the cloud IDM policy defends against actual methods hackers use to target user accounts. A centrally controlled robust cloud IDM must be in place to cover cloud services, cloud applications, outsourced IT, vendors and third parties. In addition to provisioning of passwords, enterprises must ensure that when employers change role or leave the company, their access is altered or removed. It is also important to know who has access and to what applications, combined with alerting mechanisms that can report on unusual log on activity on cloud servers.

5 Conclusion

Cloud computing is relevant in providing valuable on-demand, elastic, scalable and reliable services to customers. A lot of effort and investment is saved by users while leveraging on applications and infrastructure provided by the CSP. Identity and access management is a critical issue in cloud computing. A malicious user can exploit the identity access to gain access to user credentials and information on the cloud. There are various architectures in place to ensure a secured IDMS. A lot of effort still needs to be put in place to ensure a robust, all-encompassing IDM for cloud computing. It is recommended that more studies will be conducted on cloud IDM policies and multi-factor authentication on all Internet facing services.

References

[1] M. Ali, S. U. Khan, and A. V. Vasilakos. "Security in cloud computing: Opportunities and challenges". In: *Information Sciences* 305.Supplement C (2015), pages 357–383. ISSN: 0020-0255. DOI: https://doi.org/10.1016/j.ins.2015.01.025.

[2] E. Bertino, F. Paci, R. Ferrini, and N. Shang. "Privacy-preserving digital identity management for cloud computing." In: *IEEE Data Eng. Bull.* 32.1 (2009), pages 21–27.

[3] M. Bradford, J. B. Earp, and S. Grabski. "Centralized end-to-end identity and access management and ERP systems: A multi-case analysis using the Technology Organization Environment framework". In: *International Journal of Accounting Information Systems* 15.2 (2014), pages 149–165. ISSN: 1467-0895. DOI: https://doi.org/10.1016/j.accinf.2014.01.003.

[4] G. Dreo, M. Golling, W. Hommel, and F. Tietze. "ICEMAN: An architecture for secure federated inter-cloud identity management". In: *2013 IFIP/IEEE International Symposium on Integrated Network Management*. May 2013, pages 1207–1210.

[5] E-Guide. *How I am Has Evolved On The Cloud: The Good And The Bad*. TechTarget Publication. 2016.

[6] I. A. Gomaa and E. Abd-Elrahman. "A Novel Virtual Identity Implementation for Anonymous Communication in Cloud Environments". In: *Procedia Computer Science* 63.Supplement C (2015). The 6th International Conference on Emerging Ubiquitous Systems and Pervasive Networks (EUSPN 2015)/ The 5th International Conference on Current and Future Trends of Information and Communication Technologies in Healthcare (ICTH-2015) / Affiliated Workshops, pages 32–39. ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2015.08.309.

[7] A. Gopalakrishnan. "Cloud computing identity management". In: *SETLabs briefings* 7.7 (2009), pages 45–54.

[8] T. Green. *Gartner's top 10 security predictions*. NetworkWorld Publication. 2016.

[9] U. Habiba, A. G. Abassi, R. Masood, and M. A. Shibli. "Assessment Criteria for Cloud Identity Management Systems". In: *2013 IEEE 19th Pacific Rim International Symposium on Dependable Computing*. Dec. 2013, pages 188–195. DOI: 10.1109/PRDC.2013.39.

[10] I. Khalil, A. Khreishah, and M. Azeem. "Consolidated Identity Management System for secure mobile cloud computing". In: *Computer Networks* 65.Supplement C (2014), pages 99–110. ISSN: 1389-1286. DOI: https://doi.org/10.1016/j.comnet.2014.03.015.

[11] M. A. Leandro, T. J. Nascimento, D. R. dos Santos, C. M. Westphall, and C. B. Westphall. "Multi-tenancy authorization system with federated identity for cloud-based environments using shibboleth". In: *Proceedings of the Eleventh International Conference on Networks*. 2012, pages 88–93.

[12] J. Madden. *Cloud-based identity management services can work for your on-premises apps, too*. TechTarget Computer Weekly Publication. Nov. 2016.

[13] P. M. Mell and T. Grance. *SP 800-145. The NIST Definition of Cloud Computing*. Technical report. Gaithersburg, MD, United States, 2011.

[14] D. H. Sharma, C. Dhote, and M. M. Potey. "Identity and Access Management as Security-as-a-Service from Clouds". In: *Procedia Computer Science* 79.Supplement C (2016). Proceedings of International Conference on Communication, Computing and Virtualization (ICCCV) 2016, pages 170–174. ISSN: 1877-0509. DOI: `https://doi.org/10.1016/j.procs.2016.03.117`.

[15] T. Spencer. "Identity in the cloud". In: *Computer Fraud & Security* 2012.7 (2012), pages 19–20. ISSN: 1361-3723. DOI: `https://doi.org/10.1016/S1361-3723(12)70075-1`.

[16] M. Stihler, A. O. Santin, A. L. M. Jr., and J. d. S. Fraga. "Integral Federated Identity Management for Cloud Computing". In: *2012 5th International Conference on New Technologies, Mobility and Security (NTMS)*. May 2012, pages 1–5. DOI: `10.1109/NTMS.2012.6208751`.

[17] A. E. Youssef. "Exploring cloud computing services and applications". In: *Journal of Emerging Trends in Computing and Information Sciences* 3.6 (2012), pages 838–847.

Facilitating Policy Adherence in Federated OpenStack Clouds with Minimally Invasive Changes

Matthias Bastian, Max Plauth, and Andreas Polze

Operating Systems and Middleware Group
Hasso Plattner Institute for Digital Engineering
University of Potsdam, Germany
firstname.lastname@hpi.uni-potsdam.de

Federated cloud setups can be constituted of a confusing amount of involved parties on both sides of providers and consumers. Hence, it is important to provide a policy mechanism that does not only allow providers to express their privacy policy statements but which also enables service consumers to express their own requirements how data should be treated by providers. However, integrating policy adherence for versatile policies necessitates profound alterations in existing cloud software platforms like OpenStack.

In this work, we demonstrate a prototypical approach that facilitates policy adherence support in OpenStack with minimally invasive changes. Employing certain concepts of the Aspect-Oriented Programming (AOP) paradigm, we present a framework that is able to inject policy adherence support in arbitrary sections of the OpenStack code base. With just a few lines of code, developers may add adherence support for policy attributes, even if the policy affects multiple OpenStack services.

1 Introduction

Cloud computing has become a pervasive method for running all kinds of applications. One of the major benefits is the possibility to acquire compute, storage and networking resources whenever they are needed. Recently, federated cloud scenarios have gained notable momentum, as they provide a viable alternative to public cloud providers [9]. Federated private cloud setups can often be comprised of many involved parties, both offering and consuming services. In such an unclear environment, it is crucial to provide means that allow providers to express their privacy policy statements, but which also enables service consumers to express their own requirements how data should be treated by providers. [5]

OpenStack is a well disseminated open source cloud platform that enjoys enormous popularity for driving many private cloud setups. One of the fundamental hurdles towards integrating proactive policy adherence into OpenStack is the tremendous implementation effort, as all services affected by a policy have to be altered significantly. We address this issue by presenting the *policyextension*-framework, which uses *PolicyExtensions* to facilitate policy adherence support with minimally invasive changes to the OpenStack code base. *PolicyExtensions* share many characteristics with plug-ins, as they are not part of the original code base.

In contrast to plug-ins, *PolicyExtensions* do not rely on plug-in mechanisms but inject their code at the locations of their own choice. The infrastructure for injecting *PolicyExtensions* is provided by the *policyextension*-framework.

Our approach provides the following characteristics:

- Interpretation of policies and the associated adherence mechanisms can be implemented in one single location, rather than being spread across the code base of numerous OpenStack services.
- Integrating policy support can be minimally invasive regarding code changes in existing services.
- Policy support code is easy to maintain.
- Facilities can be easily extended in order to support additional policy attributes.

The remainder of the paper is structured as follows: Section 2 provides an overview of prior approaches for providing policy adherence support in OpenStack. Furthermore, the employed policy language *CPPL* [6] is introduced. Afterwards, Section 3 elaborates on the fundamental design decisions that lead to the approach presented in Section 4. In Section 5, the applicability of the approach is demonstrated. Finally, Section 5 concludes this work.

2 Related Work

In this section, we provide an overview of prior policy integration concepts in OpenStack. We further introduce the policy language which is employed by the policy adherence mechanisms presented in this paper.

2.1 Policy Support in OpenStack

Here, we provide a brief overview of approaches for supporting policies in OpenStack that existed prior to this work.

2.1.1 oslo.policy

The *Oslo* project offers a plentitude of infrastructure utilities, e.g. for facilitating database access or for providing message queues. Also part of the project, *oslo.policy* is a dedicated library for managing inter-project communication. The library defines a format for specifying rules and policies and provides a corresponding policy execution engine. However, this policy engine does not suffice the requirements of federated private cloud setups, as *oslo.policy* is mainly intended to be used for authorization purposes. Potentially, *oslo.policy* can be used to guard other request properties for which *permit-or-deny* semantics are sufficient. However, more complex policies ought to be supported.

2.1.2 Swift Storage Policies

Swift is the OpenStack project for providing object storage. Policy support is provided at the fundamental level, as the containers holding objects can be annotated with policies such as replication rate. Policies are defined by users during the creation of a container, however policies are restricted to core concerns of the object storage and may not be used by other OpenStack projects.

2.1.3 Policy-based Scheduling in Nova

Upon creation of a new virtual machine, OpenStack has to decide on which host the VM should be instantiated. The *Nova* scheduler therefore attempts to locate a host that fulfills several predefined policies. Using this policy-based scheduling approach enables restricting VM instantiation requests to certain hosts [7]. Custom policies can be added by OpenStack instance operators by extending the policy set employed by the *Nova* scheduler. The major disadvantage of this approach is that users can neither review nor edit the policies that are applied to requests. Only OpenStack operators are able to add, review or edit policies. Another shortcoming of this approach is that the policy support is restricted to the *Nova* project.

2.1.4 Congress

The *Congress* project is a dedicated OpenStack project that aims at providing a centralized policy component for enabling compliance in cloud-based environments. Congress employs a monitoring approach in order to maintain a high degree of independence among OpenStack projects. It detects policy violations in a passive mode of operation by querying the state of all involved OpenStack services in regular intervals using their corresponding APIs. In case the state of an OpenStack service deviates from a policy, the violation is logged and notifications can be triggered.

One major limitation of *Congress* is that its monitoring-based approach impedes the implementation of proactive adherence mechanisms. Policies may specify how violations should be treated. In addition to logging violations and triggering notifications, policies can also be configured to revert policy violations. However, several actions are hard to revert, especially in cases where the violating action triggers many side-effects that have to be reverted as well. Potentially, proactive policy adherence based on *Congress* can be implemented. However, this would necessitate significant changes to the code base of all involved OpenStack services.

2.2 Compact Privacy Policy Language (CPPL)

In the context of the *Scalable and Secure Infrastructures for Cloud Operations* (SSICLOPS) project[1], many policy languages, including XACML [4], C^2L [13] and S4P [2], have been examined with regard to their applicability in federated cloud computing scenarios. [3] This survey revealed that none of the evaluated approaches

[1] `https://ssiclops.eu`

sufficiently satisfied the extensive list of assessment criteria. In consequence of these findings, the *Compact Privacy Policy Language* (CPPL) [6] has been designed to address expressiveness and extensibility, but at the same time *CPPL* addresses the mostly neglected requirements of runtime performance as well as communication and storage efficiency. In the scope of this work, one key attribute of *CPPL* is the very compact binary representation of its policy annotations, which facilitates efficient processing and enables the use of policy annotations at a per data-item level.

3 Design Considerations

In this section, we provide a brief discussion of the most crucial aspects that strongly influenced the design of the architecture presented in Section 4 for integrating policy support in OpenStack.

3.0.1 Monitoring versus Proactive Adherence

As elaborated in the context of *Congress* (see Section 2.1.4), proactive adherence to policies requires numerous changes in the OpenStack code base compared to a monitoring-based approach. However, the proactive approach never allows policy violations to occur in the first place, whereas monitoring-based approaches are limited to reacting to policy violations by means of logging and issuing counteracting actions. Here, we aim for the proactive approach.

3.0.2 Versatility of Policies

The *SSICLOPS* policy language *CPPL* (see Section 2.2) supports a wide range of policy attributes. Due to this versatility of *CPPL*, policies can affect any OpenStack project. In order to deal with this high degree of versatility, each OpenStack service might have to be adapted in order to support *CPPL*. As a result, an implementation strategy is required that allows for the integration of policy support with minimally invasive changes to the OpenStack code base.

3.0.3 Development Process of OpenStack

OpenStack projects are strictly separated in order to prevent inter-service dependencies. This level of isolation is also reflected by the development process: Projects may only interact using their regular, public APIs. Contributing code to OpenStack projects involves a very complex workflow [10] [12], which goes far beyond the usual habits of the typical fork-merge workflow applied on many GitHub projects. As this process introduces a fair amount of complexity, it is not feasible to perform changes across many OpenStack projects, as it would be necessary in order to implement adherence even to simple policies. Therefore, a central requirement for the presented approach is to keep the overhead for implementing policies as low as possible.

4 Approach

Our approach is comprised of two major components, the *policymiddleware* and the *policyextension*-framework. Both are documented hereinafter.

4.1 *policymiddleware*-Component

One of the central concepts of *CPPL* is, that policy annotation occurs at a per data-item level. Hence, any OpenStack component needs to be able to access policy information in order to further process it. Extending the OpenStack APIs with policy support is not an option, as it contradicts the design goal of implementing policy support with minimally invasive changes to the OpenStack code base. The effort required for adapting higher level APIs is manageable, however since higher level requests (e.g. to the *Horizon* dashboard) trigger many subsequent requests to lower level APIs, the consequent necessity for code changes affects the entire OpenStack code base.

In order to make policy information transparently available to arbitrary OpenStack components, we introduce a new middleware component called *policymiddleware*, which is based on the general concepts of the *keystonemiddleware* component. To unburden other OpenStack components of the task of user authentication, all requests pass through the *keystonemiddleware*, which verifies the validity of the `X-Auth-Token` in the HTTP headers first. Requests with invalid authentication tokens are rejected right away and never reach the intended service, whereas valid requests are annotated with user credentials such as username and the user identifier. In a similar fashion, the *policymiddleware* validates incoming *CPPL* annotations and deposits the policy information in *Keystone*, from where it can be retrieved from the *policyextension*-framework. In cases where incoming requests are not annotated with policies in the first place, the *policymiddleware* fetches user-based policy information from *Keystone* and annotates the request with the resolved policies. The mechanics of both the *keystonemiddleware* and the *policymiddleware* are illustrated in Figure 1.

4.2 *policyextension*-Framework

With the *policymiddleware* providing policy information, the logic for interpreting and adhering to policies is still required. To close this gap, we introduce the *policyextension*-framework, which enables developers to implement policy support through so-called *PolicyExtensions*, which share certain properties with plug-ins. Unlike plug-ins, however, they do not rely on potentially missing extension facilities but rather inject their logic through *monkey patching* mechanisms, where the framework modifies the behavior of selected classes and functions at runtime. Even though monkey patching is often considered to be an obscure method, we employed it anyway to compensate for the lack of a unified plug-in infrastructure across OpenStack projects. Furthermore, with proper safeguards like version checking in place,

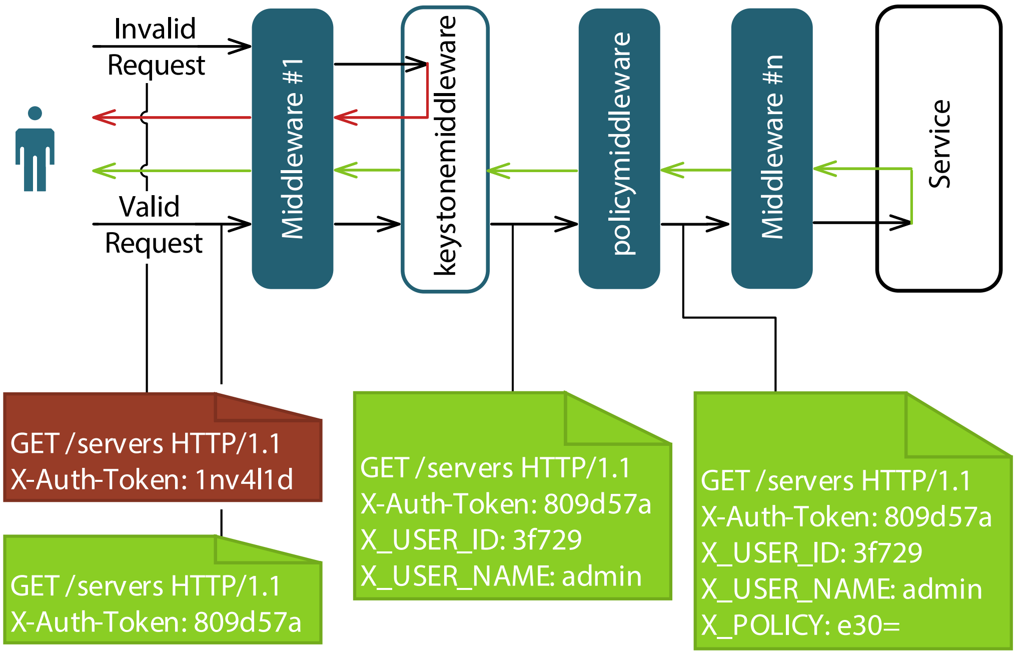

Figure 1: Analogous to *keystonemiddleware*, the *policymiddleware* component transparently annotates requests with policy information.

monkey patching is even used in production code. The *Nova* project for example employs *monkey patching* indirectly by employing the *eventlet* library [11].

Developers can implement new *PolicyExtensions* by creating a new class that inherits from the base class `PolicyExtensionBase`. In this new class, the attribute `func_paths` has to be provided, which should contain a list of functions to be modified. Furthermore, a method with the same name has to be implemented, containing the logic that is executed prior to the execution of the original function. The *policyextension*-framework handles the entire patching process, which is visualized in Figure 2. It furthermore provides access to the arguments that the original function is called with. As the policy information is initially only available at the API endpoints of each service, the framework also makes this information available from arbitrary locations in the service implementations. Last but not least, the framework converts incoming policy information to *Python*-friendly dictionaries, which can be easily queried. Since this format conversion mechanism employs a well-defined interface for decoding and encoding policy information, support for other policy annotation languages can be easily retrofitted.

The mechanisms provided by the *policyextension*-framework implement many concepts of the *Aspect Oriented Programming* (AOP) paradigm [8]. Mapping the components of the presented framework to AOP concepts, `PolicyExtension` classes resemble *Aspects* whereas the original functions to be modified correspond to *Jointpoints* and the `func_paths` represents the *Pointcut*. The method of a `PolicyExtension` class corresponds to an *Advice*.

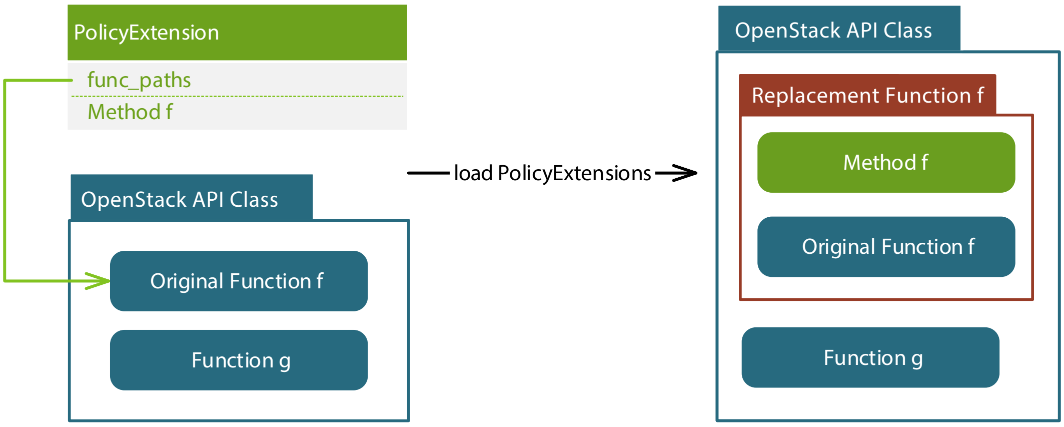

Figure 2: *PolicyExtensions* dynamically modify the behavior of OpenStack classes and functions at runtime by applying *monkey patching*.

5 Applicability

Here, we demonstrate the applicability of the approach presented in Section 4 by providing example code for supporting two policy attributes.

5.1 Enforcing Disk Encryption

The *policyextension*-framework uses the dictionary data type `dict` in *Python* to provide the policy information. Listing 1 shows the `dict`-based representation of a policy that requires the use of disk encryption.

Listing 1: `dict`-based policy notation for enforcing disk encryption:

```
{
  "storage": {
    "encryption": True
  }
}
```

To support adherence to such a policy, the *Cinder* service of OpenStack has to be adapted. In OpenStack, the Cinder service is responsible for providing *Block Storage*, which is indicated by the `storage` key. The embedded key `encryption` with the corresponding boolean value `true` then specifies, that newly created volumes must use encryption. Below, the example code demonstrates how the proactive policy adherence can be implemented by creating a new *PolicyExtension*:

Listing 2: Example *PolicyExtension* for enforcing disk encryption policies:

```
from policyextension import PolicyExtensionBase, PolicyViolation
from cinder.volume import volume_types

class CinderEncryptedVolumeTypeRequiredExtension(PolicyExtensionBase):
  func_paths = ['cinder.volume.api.API.create']

  def create(self, func_args, policy):
    try:
      if policy['storage']['encryption']:
        volume_type = func_args['volume_type'] or volume_types.get_default_volume_type()
        if not volume_type or not volume_types.is_encrypted(func_args['context'], volume_type['id']):
          msg = "Your policy requires using an encrypted volume type."
          raise PolicyViolation(msg)
    except KeyError:
      pass
```

5.2 Restriction of Availability Zones

As a second example, we demonstrate the code for supporting adherence to a policy that restricts the instantiation of virtual machines to a set of whitelisted availability zones. Here, we are using the OpenStack mechanism of availability zones in order to model geographic locations. Under the assumption that an *Availability Zone* with the name az2 exists, the human readable representation of the policy for this example is demonstrated in Listing 3.

Listing 3: dict-based policy notation for restricting availability zones:

```
{
  "availability_zones": ["az2"]
}
```

Since we are using availability zones to model geographic locations of the data center, we decided to employ a whitelist approach in order to ensure that services are only instantiated in a region that is explicitly approved by the user. The resulting implementation is presented in Listing 4.

Listing 4: Example *PolicyExtension* that can restrict the execution of virtual machine instances to whitelisted availability zones:

```
from policyextension import PolicyExtensionBase, PolicyViolation
import random

class AvailabilityZoneRestrictionExtension(PolicyExtensionBase):
  func_paths = ['nova.compute.api.API.create']

  def create(self, func_args, policy):
    availability_zone = func_args['availability_zone']
    try:
      az_whitelist = policy['availability_zones']
      if availability_zone:
        if availability_zone not in az_whitelist:
          msg = ("Your policy does not allow the availability zone you selected.")
          raise PolicyViolation(msg)
      elif az_whitelist:
        func_args['availability_zone'] = random.choice(az_whitelist)
    except KeyError:
      pass
```

6 Conclusion

In this paper, we presented the *policyextension*-framework, which uses *PolicyExtensions* to facilitate policy adherence support in OpenStack with minimally invasive changes to the code base. To demonstrate the applicability of the framework, we discussed exemplary implementations that provide policy adherence support for enforcing disk encryption and for restricting virtual machines to whitelisted availability zones. Both implementations employ *permit-or-deny* semantics. As next steps, we would like to evaluate modifying semantics, where incoming requests are modified in order to satisfy policy requirements. Furthermore, we are planning to evaluate our approach in a federated OpenStack testbed.

All results are based on the master's thesis by Matthias Bastian. Additional implementation details have been documented in the original thesis [1].

Acknowledgement & Disclaimer

This paper has received funding from the European Union's Horizon 2020 research and innovation programme 2014-2018 under grant agreement No. 644866. This paper reflects only the authors' views and the European Commission is not responsible for any use that may be made of the information it contains.

References

[1] M. Bastian. "Design and Integration of a Framework for Enforcing User-Defined Policies in OpenStack". Master's Thesis (in German). Hasso Plattner Institute for Software Systems Engineering, University of Potsdam, Jan. 2017, page 88.

[2] M. Y. Becker, A. Malkis, and L. Bussard. "A Practical Generic Privacy Language". English. In: *Information Systems Security*. Edited by S. Jha and A. Mathuria. Volume 6503. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2010, pages 125–139. ISBN: 978-3-642-17713-2. DOI: 10.1007/978-3-642-17714-9_10.

[3] F. Eberhardt, M. Plauth, A. Polze, S. Klauck, M. Uflacker, J. Hiller, O. Hohlfeld, and K. Wehrle. *SSICLOPS Deliverable 2.1: Report on Body of Knowledge in Secure Cloud Data Storage*. Technical report. June 2015.

[4] S. Godik, A. Anderson, B. Parducci, P. Humenn, and S. Vajjhala. *OASIS eXtensible Access Control Markup language (XACML)*. Technical report. OASIS, May 2002.

[5] M. Henze, M. Großfengels, M. Koprowski, and K. Wehrle. “Towards Data Handling Requirements-Aware Cloud Computing”. In: *2013 IEEE 5th International Conference on Cloud Computing Technology and Science*. Volume 2. Dec. 2013, pages 266–269. DOI: 10.1109/CloudCom.2013.145.

[6] M. Henze, J. Hiller, S. Schmerling, J. H. Ziegeldorf, and K. Wehrle. “CPPL: Compact Privacy Policy Language”. In: *Proceedings of the 2016 ACM on Workshop on Privacy in the Electronic Society*. WPES ’16. Vienna, Austria: ACM, 2016, pages 99–110. ISBN: 978-1-4503-4569-9. DOI: 10.1145/2994620.2994627.

[7] Khanh-Toan Tran and Jérôme Gallard. *A new mechanism for nova-scheduler: Policy-based Scheduling*. Technical report. 2013.

[8] G. Kiczales, J. Lamping, A. Mendhekar, C. Maeda, C. Lopes, J.-M. Loingtier, and J. Irwin. “Aspect-oriented programming”. In: *ECOOP’97 – Object-Oriented Programming: 11th European Conference Jyväskylä, Finland, June 9–13, 1997 Proceedings*. Edited by M. Akşit and S. Matsuoka. Berlin, Heidelberg: Springer, 1997, pages 220–242. ISBN: 978-3-540-69127-3. DOI: 10.1007/BFb0053381.

[9] R. Moreno-Vozmediano, R. S. Montero, and I. M. Llorente. “Iaas Cloud Architecture: From Virtualized Datacenters to Federated Cloud Infrastructures”. In: *Computer* 45.12 (2012), pages 65–72.

[10] OpenStack Foundation. *Developer’s Guide*.

[11] OpenStack Foundation. *Nova Threading Model*.

[12] OpenStack Foundation. *OpenStack Blueprints*.

[13] J. Poroor and B. Jayaraman. “C2L:A Formal Policy Language for Secure Cloud Configurations”. In: *Procedia Computer Science* 10 (2012). {ANT} 2012 and MobiWIS 2012, pages 499–506. ISSN: 1877-0509. DOI: http://dx.doi.org/10.1016/j.procs.2012.06.064.

Dependability Stress Testing Through Model-based Fault Injection

Keynote

Lukas Pirl

Operating Systems and Middleware Group
Hasso Plattner Institute for Digital Engineering
University of Potsdam, Germany
lukas.pirl@hpi.uni-potsdam.de

In the context of complex, fast-evolving, distributed systems, the approach of software fault injection for experimental dependability assessments does not seem to be unfolded to its full potential yet. We propose a structured method to derive software fault injection campaigns from user-provided dependability models. Such a campaign tests all combinations of as many concurrently tolerable faults as possible (i.e., "dependability stress") and, thus, tests for synergistic effects. Additionally, we present a flexible framework to automate the aforementioned derivation and to coordinate the campaigns' exercise. In a case study, we assess the dependability of a framework to build IaaS platforms accordingly. Finding an adequate granularity for the dependability model, setting up a complex distributed system virtualized and automating its restoration from snapshots are especially challenging aspects. However, experimental, structured and continuous dependability assessments are considered crucial for the functioning of distributed systems, which society increasingly depends on.

Cloud and E-Learning – Current Issues and Developments

Isaac Odun-Ayo, Sanjay Misra, Nicholas Omoregbe, and Ambrose Azeta

Department of Computer and Information Sciences
Covenant University, Nigeria
firstname.lastname@covenantuniversity.edu.ng

Cloud computing is a revolutionary IT paradigm that is dynamically providing useful services to IT consumers. The cloud has providers that make software available for users to carry out basic routine tasks. Cloud providers have also provided a platform for users to design, deploy and run applications on the cloud. In addition, storage and computing power is also being provided on the cloud. These provisions available on the cloud makes it most suitable for E-Learning purposes. E-Learning provides a web-based forum for instructors and students to meet. Both hardware and software systems are required for this purpose. This can easily be implemented using available cloud infrastructure. This paper examines present trends in the area of cloud E-Learning and provides a guide for future research. The study was executed by means of review of some literature available on cloud E-Learning. Papers published in journals, conferences, white papers and those published in reputable magazines were analyzed. The expected results will be of immense benefit to prospective cloud users and cloud providers.

1 Introduction

"Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction" [10]. E-Learning involves all manner of learning and teaching in an electronic form. The information system could be a networked type or stand alone, but it serves as the medium through which the learning process is carried out. The E-Learning process could be carried in classrooms or outside the class using the information and communication system which continues to advance on a regular basis. E-Learning uses computers, networks and the Internet to provide education, knowledge and skills to relevant persons. There are various media through which E-Learning information is transferred such as using the Internet, audio and video CDs, local and satellite TV broadcast. The E-Learning process can be done on an individual basis or through an instructor through audio, video, text and animations.

The E-Learning process mainly utilizes the Internet, using relevant technologies to design, develop, implement and maintain the learning environment [8] [13]. Although the E-Learning process has greatly improved learning, it is not expected to completely replace traditional methods of learning. E-Learning has various applications that are implemented using the web, the internet, computer-enabled environment and various virtual and collaborative means [11] [9] [8]. E-Learning is widely used for distant training, organizational training and educational applications. E-Learning applications are available both on commercial or proprietary basis and open source. Advances in E-Learning has made it possible to have relevant standards for both content and components. Such standards include the IEELOM, UKLOM, IMS, SCORM and OKI [9]. The Dublin core E-Learning standard provides facilities to optimize electronic resources, institutions of higher learning and other educational collaborations. The Dublin core metadata allows for the development of market-based standards which are open for online purposes including specifications [9].

Learning content can be hosted on several platforms, such as web servers, cloud servers, among others. Cloud computing provides a mutually beneficial springboard for cloud providers and users to enhance IT service delivery. Cloud computing provides scalable, elastic and on demand services that can be hosted on the Internet. Cloud computing is highly scalable and provides virtualized resources that can be made available to users [11]. Cloud service providers (CSP) utilizing data centers with state-of-the-art technology, offer IT services to users at a cost. The cloud infrastructure has compute and storage capacity and it's also a platform for hosting applications. There are three primary services hosted on the cloud; Software-as-a-Service (SaaS), Platform-as-a-Services (PaaS) and Infrastructure–as-a-Service (IaaS). SaaS is a software distribution model, through web delivery, enabling users to access applications on the Internet [9]. Software applications are offered as a service over the Internet instead of users purchasing licensed software. PaaS provides facilities to support the entire application development life cycle and the deployment of such applications on the platform provided by the CSP [8].

The user is able to create and deploy an application using the APIs provided by the CSP. This frees the user from the need for an operating system or other computing infrastructure. In IaaS, storage and computer resources through the process of virtualization and multi-tenancy, is provided to the user. For those with existing data centers, auxiliary activities can be migrated, while start-ups do not require the expensive IT infrastructure for operations. Clouds can be deployed in four primary ways. There are private, public, community and hybrid clouds. Private clouds are owned by individual organizations with full control over the entire infrastructure. The facility can be on premise or off premises and may be managed by a third party. It is considered more secure than other deployment types. Public clouds are owned by major cloud providers who operate extensive and expensive data centers across geographical locations. They provide users with storage, compute and other services on demand and on a pay-as-you-go basis. Community cloud is usually hosted by several organizations with a shared common interest. It is either managed by the community or a third party. Several organization share infrastructure based on some policy regarding security, compliance and other issues [13]. A hybrid cloud is a combination of private, public or community cloud. Hybrid clouds remain a unique entity but enjoy the benefits of the constituent deployment types.

Cloud computing and E-Learning is a subdivision of cloud computing in education. E-Learning based on the cloud makes use of cloud hardware and software infrastructure to enhance the traditional methods of learning. The E-Learning educational materials are ported to the cloud and placed in a virtual environment by the designers or service providers and thereafter made available to the users. The unique characteristics of cloud computing has provided the relevant infrastructure to E-Learning, making it valuable to students, teachers and researchers alike. Applying Cloud computing allows the E-Learning process to focus on teaching rather than on complex IT hardware and software management. There are numerous benefits which cloud computing offers to E-Learning in terms of storage, computing power, virtualization and multi-tenancy which is relevant to offering educational services [8].

The aim of this paper is to discuss the process of Cloud based E-Learning. The paper provides an insight into E-Learning and highlights events currently going on in the industry to enhance E-Learning. The remaining part of the paper is organized as follows: Section 2 examines related work. Section 3 discusses the benefits of cloud computing in E-Learning. Section 4 discusses the architecture of E-Learning in relation to Cloud computing. Section 5 highlights current trends by examining Hochschule Furtwangen University (HFU). Section 5 concludes the paper and makes future recommendation.

2 Related Work

E-Learning using cloud computing in [3] examines the benefits of using the cloud for E-Learning. Several institutions cannot afford the huge investments involved in E-Learning making the cloud a viable option. Measuring the efficiency of cloud computing for E-Learning systems in [11], observes that E-Learning requires huge investments in hardware and software. The cloud becomes a useful tool to save cost. The paper examines the efficiency of E-Learning on cloud computing based on some metrics. A novel approach for a adopting cloud-based E-Learning system in [8] proposed steps that institutions could follow in adopting cloud computing for E-Learning. Two subsystems were proposed for the utilization of the E-Learning cloud. An adaptive framework for applying cloud computing in virtual learning environment at education: a case study of Arab Academy for Science, Technology & Maritime Transport (AASTMT) in [6] proposes a model for implementing E-Learning on the cloud. The framework was implemented and feedback showed a favorable outcome.

An E-Learning system architecture based on cloud computing in [9] observed that information and communications technology (ICT) is enhancing education. An E-Learning cloud was created using certain metrics and the benefits were also discussed. Antecedents and consequences of cloud computing adoption in education to achieve knowledge management in [2] implements knowledge management practices in cloud computing. The model was validated using a survey and the conclusion was on the need for increased awareness. A perspective on E-Learning and cloud computing in [12] discusses E-Learning from the perspective of virtual laboratories. A cloud virtual laboratory system is proposed to improve learning using the cloud infrastructure. Comparative Analysis for Cloud-based E-Learning in [1] examined cloud services and E-Learning standards. The main focus was to conduct an analysis of E-Learning standards in the traditional and cloud forms. Instance-based ontology matching for E-Learning material using associative pattern classifiers in [4] uses a unique pattern algorithm for classification with a view to improve resources for E-Learning.

E-Learning systems based on cloud computing: a review in [5] observed that it has become essential to utilize the cloud with a view to providing E-Learning services. The availability of efficient cloud infrastructure will ensure that both the cloud providers and consumers benefit from E-Learning on the cloud. The design virtual learning labs for courses in computational science with use of Cloud computing technologies in [5] discusses the virtual lab using application as a service. Features of the model were discussed and implemented. The result will enhance teacher ability to evaluate student's understanding. Cloud computing for E-Learning in [7] highlights the services available on the cloud and also discusses the benefits of using the cloud for E-Learning.

3 Benefits of Cloud Computing in E-Learning

Cloud computing has the following benefits on E-Learning [11] [13] [1] [5]:

- **Infrastructure:** The use of hardware and software solutions based on the E-Learning infrastructure provisioned by the CSP. Hence, the use of an E-Learning solution on the providers infrastructure to facilitate the process of E-Learning.

- **Services:** The use of an E-Learning solution provided in SaaS by the service provider.

- **Content:** The use and development of an E-Learning solution based on platform provided by the CSP through PaaS.

- **Low Cost:** Those using E-Learning applications do not need to have state-of-the-art systems to connect to E-Learning resources. Cloud-based E-Learning can be done easily through mobile phones and PCs with relevant connections to the Internet.

- **Improved Performance:** In the application of SaaS, the CSP has control of software and infrastructure including maintenance. Since most applications and processes related to cloud-based E-Learning are resident on the cloud, there is improved performance in the E-Learning process.

- **Regular Software Update:** A Cloud user operating an E-Learning platform does not need to bother about software updates and other maintenance issues.

- **Improved Document Format Compatibility:** Some documents do not open on certain applications due to format incompatibility. An E-Learning application or other software will resolve this issue on the Cloud through the CSP.

- **Benefits for Students:** Cloud application are hosted on the Internet and can be accessed anywhere at any time simultaneously. This implies that students can take online courses, online examinations, send their projects and assignments to their instructors and also get feedback on all these things.

- **High Computing Storage Capacity:** Cloud based E-Learning does computing and stores data in large data centers with geographical spread providing huge computing power and storage for E-Learning purposes.

- **High Availability of Services:** Cloud computing provides on demand and scalable services that is available at all time in several places to users, enhancing the effectiveness of E-Learning.

- **Benefits for Teachers:** Teachers involved in E-Learning also enjoy numerous benefits. They can interact with students, prepare online courses and also examine students.

4 E-Learning Abstractions Architecture

The E-Learning cloud is a unique cloud technology that allows for all the software and hardware components to develop the E-Learning environment in a futuristic manner. By utilizing the cloud characteristics of virtualization, the educational materials can be made available to students, teachers and researchers [11] [8] [13] [1]. The E-Learning architecture is shown in Figure 1. As shown in Figure 1, the user is exposed to E-Learning resources on the E-Learning cloud.

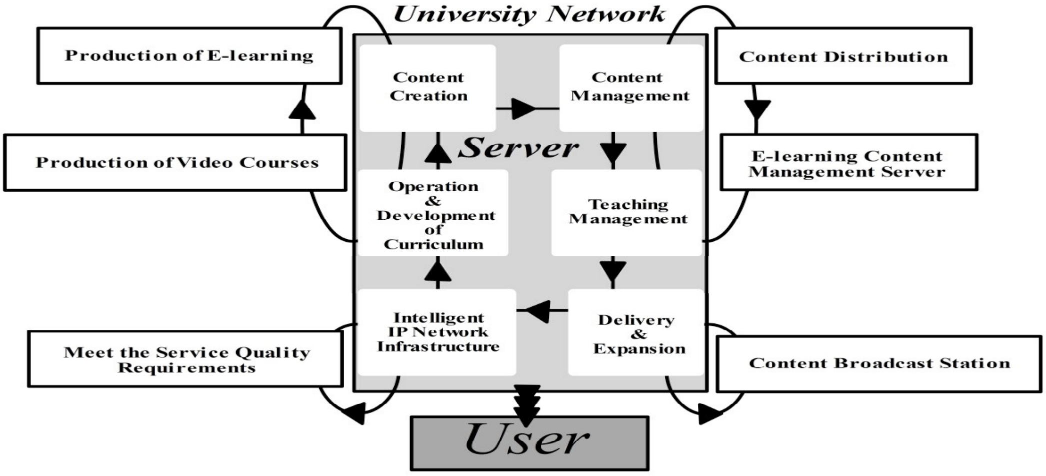
University Network
Production of E-learning
Production of Video Courses
Content Creation
Content Management
Server
Operation & Development of Curriculum
Teaching Management
Intelligent IP Network Infrastructure
Delivery & Expansion
Content Distribution
E-learning Content Management Server
Meet the Service Quality Requirements
Content Broadcast Station
User

Figure 1: E-Learning Architecture [13]

The basic E-Learning is new to education, utilizing IT resources. On the other hand, cloud-based E-Learning provides a unique opportunity to students, faculty and researcher in any academic environment because of the benefits provided by the cloud infrastructure. The students can connect to the E-Learning resources anywhere on campus to obtain study materials. Faculty members can also take advantage of the E-Learning cloud to prepare lecture notes and also interact with the students. Finally, the research community has access to a virtualized infrastructure with adequate network bandwidth and compute power for research purposes. The services provided by an E-Learning cloud is shown in Figure 2 [13].

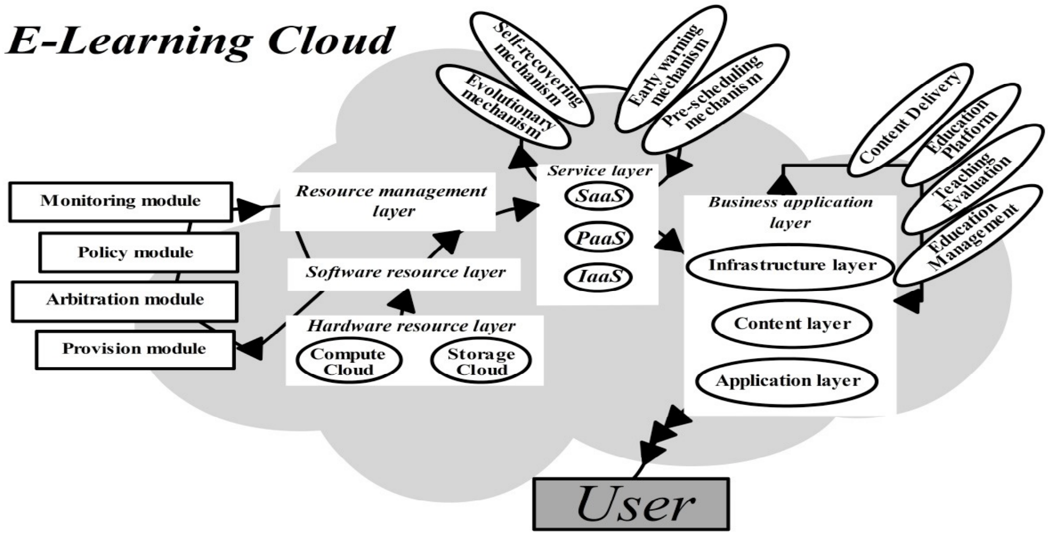

Figure 2: Services Provided By an E-Learning Cloud [13]

The Cloud computing E-Learning architecture is divided into five layers, namely: infrastructure, software, resource management, service and application. This is depicted in Figure 3 [11] [9] [1].

Infrastructure Layer This layer provides the teaching and information infrastructure. This includes the application software, the internet, information system and others. The teaching materials are obtained from the traditional methods and transferred to the cloud environment. In the Cloud services middleware, the infrastructure layer is the lowest layer. The computing resources are also available in this layer. Virtualization technology enable the storage, servers and the network to be accessed by other levels. The physical host pool is scalable and dynamic, hence, new instances are easily provisioned to improve the computing power provided by the middleware. The infrastructure layer is depicted in Figure 3.

There is also a monitoring module. The module keeps track of execution of request, real time configuration information and resource utilization.

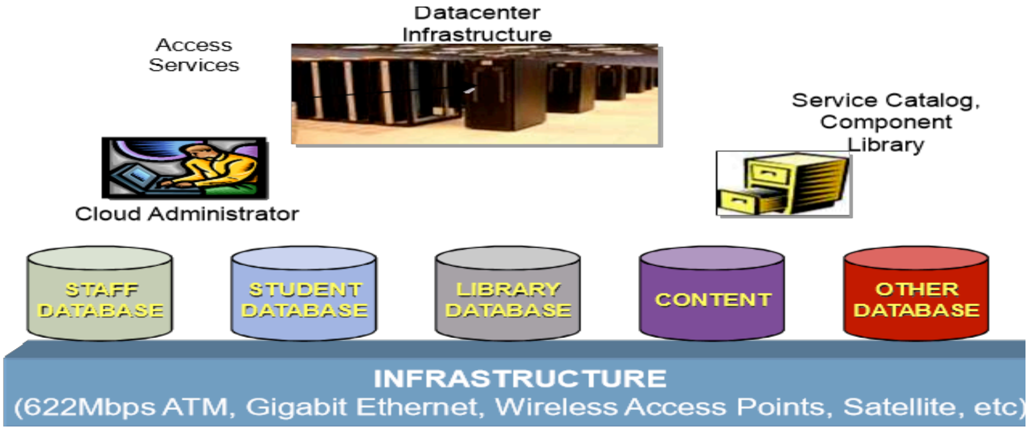
Datacenter Infrastructure
Access Services
Cloud Administrator
Service Catalog, Component Library
STAFF DATABASE
STUDENT DATABASE
LIBRARY DATABASE
CONTENT
OTHER DATABASE
INFRASTRUCTURE
(622Mbps ATM, Gigabit Ethernet, Wireless Access Points, Satellite, etc)

Figure 3: The infrastructure layer [9]

Software Layer The software layer mainly consists of the application and system software making up the middleware. Based on the technology of the middleware, various resources are made available to allow developers design, develop and deploy applications and make such applications available to users.

Resource Management Layer This layer is relevant for ensuring necessary interface between the hardware and software. Based on virtualization, schedule strategy, on-demand usage, the free flow and distribution of software over the various hardware resources can be achieved. The modules associated with this layer are as follows:

- **Policy module:** the policy module establishes a framework for teaching, learning, resource scheduling and for the runtime.
- **Arbitration module:** the arbitration module is used to make policies, complete user request and resolve and conflicting resource issues.
- **Provision module:** this module initiates the allocation of resources based on collaboration with the processes in the preceding modules.

Service Layer The service layer provides the services offered by the three primary service types in Cloud computing. This layer enables cloud users to utilize the various cloud resources available for their products.

Business Application Layer The business layer allows the coordination of teaching resources on the cloud. This is done through sharing of resources and other interactive processes. The interactive process is meant for faculty members based on the teaching and learning requirements of the student. Adjustments can be made from feedbacks based on the underlying resources and the progress of learning. This layer consists primarily of content production, educational objectives content delivery technology, and assessment and management component.

Virtualization Layer The virtualization layer deals with the virtual machines utilized in the cloud model. Cloud resources such as servers, storage and networks are provisioned in an elastic manner using visualization to ensure that the cloud E-Learning process offers the best services. Virtualization is a hardware feature that allows multiple operating systems to run concurrently on a host computer.

5 The Hochschule Furtwangen University E-Learning Cloud Architecture

5.1 The HFU E-Learning Infrastructure

The Hochschule Furtwangen University (HFU) [14] provides a good example of the application of cloud E-Learning. HFU is a typical university scenario, where PC labs and servers are underutilized at night and during semester breaks. Cloud computing can be used to complement E-Learning activities in such an environment. As a result HFU established the cloud infrastructure and application. The main objective of the HFU project is to provide a cloud of a private type with an elastic, on-demand access for E-Learning process through a Servlet Container Platform and on-demand collaborations software. Figure 4 shows the HFU private cloud type that was developed on an already existing facility. It comprises computer pools of 3 types: PC pool, research pool and server pool.

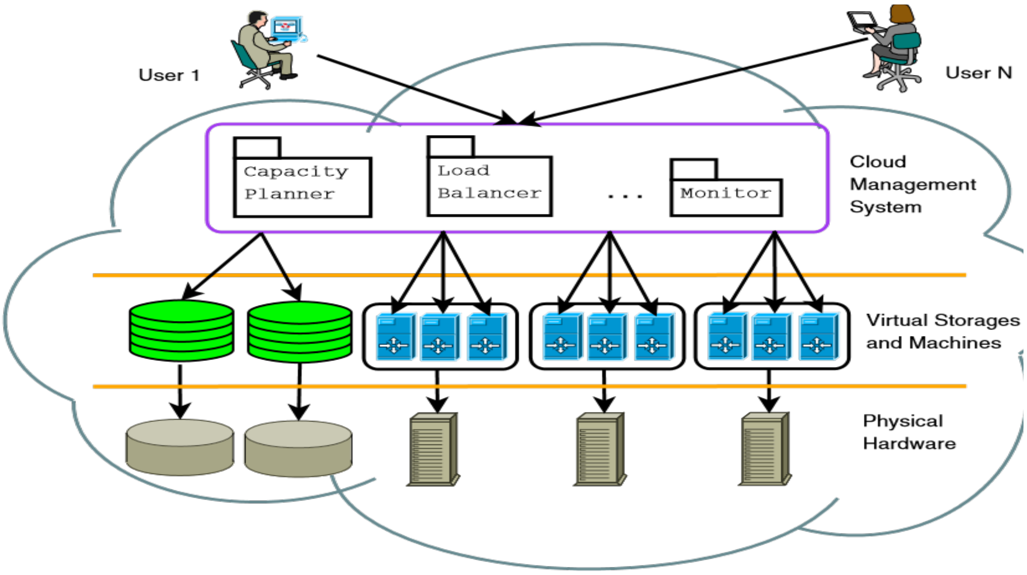

Figure 4: Overview of the CloudIA architecture [14]

The PC pool has 18 computers with Ubuntu Operating System and KVM, while other pools have Debian OS and Xen hypervisor configuration. The PC and server pools are utilized by faculty and students for teaching and learning purposes, while research and development is done on the research pool. Pools management is done through CloudIA's Cloud Management System (CMS) as shown in Figure 5.

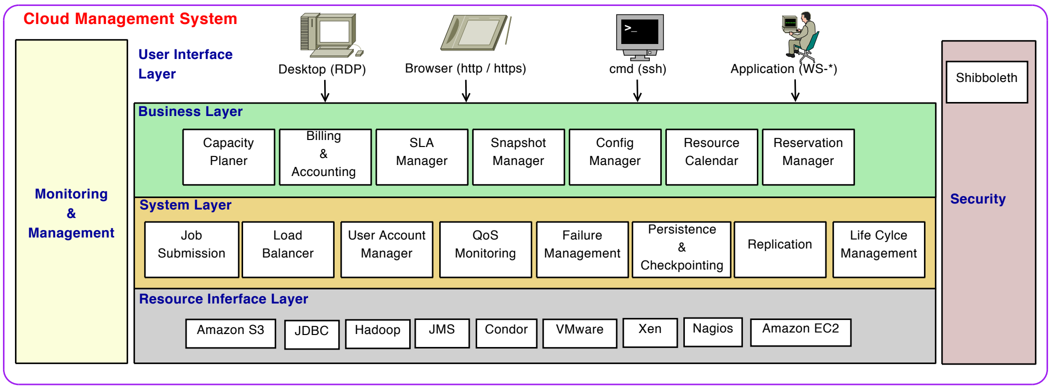

Figure 5: Cloud Management System of CloudIA [14]

The CMS is broken into various layers to allow for scalability and maintenance. A security, management and monitoring component is available in all the layers to ensure efficient services. The description of the layers is as follows:

- **User Interface Layer:** this is like an access layer for all users of the CMS users, enabling both the users and administrators alike to access the system.
- **Business Layer:** this layer deals with the economics of cloud computing by allowing price and service level agreements issues to be optimized. It also allows virtual machines to be reserved by users in advance, including control of the virtual machines (VMs).
- **System Layer:** this is an operational layer that provides the environment to optimize the quality of service, job queues and management of user accounts.
- **Resource Interface Layer:** this is the layer responsible for the hardware. Relevant interfaces are provided to connect to the databases, the virtual environment and other systems such as Xen, Amazon EC2, Amazon S3 and Nagios.
- **Monitoring and Management Component:** this layer allows for adequate management and monitoring of all the other layers to ensure reliability of the system. This system administrator is able to mitigate against likely failures, resolve SLA issues and optimize the utilization of resources.
- **Security Component:** to ensure recovery privacy, integrity and security of user data and transaction, a security layer is provided.

5.2 The HFU and Single Sign on Using Shibboleth

To prevent duplicating user management, the HFU system utilizes the single sign on using (SSO) Shibboleth for authentication purposes. This includes access to the CloudIA platform services such as servlet container Platform (SCP) and CollabSoft. Shibboleth provides the needed interface between the existing infrastructure and the HFU cloud services. Shibboleth's primary tools are the identity provider (IdP), service provider and discovery service for localization of the IdP. Federation is simple to provide because a user can process authentication through the home IdP, while the authorization is also granted through the service provider of the same home IdP. Shibboleth provides access to the CloudIA platform. A major requirement of Shibboleth is that all service providers produce a certificate that is valid. Based on the fact that the certificates are processed dynamically, the certificate of an instance is available at runtime. Consequently, the CloudIA has a record of all pre-set certificates for the service providers.

Virtual Machine Creation There is a front-end in the system which allows students to log in, create, delete or suspend virtual machines based on their needs using a wizard. A student is provided with a maximum of three VMs and a maximum of 1Gb per VM with one hundred hours of CPU time per semester. There are two unique features: creator and monitoring of VMs in CloudIA. Also, users can decide on the type of applications they want to install during the VM creation, instead of the usual preconfigured VM images.

Platform-as-a-Service: Servlet Container Platform A servlet container platform (SCP) is implemented using PaaS for HFU courses. The courses include middleware such as Java frameworks and web frameworks. The SCP enables students to design and deploy their applications to the E-Learning infrastructure without having to install and configure software as shown in Figure 6.

Software-as-a-Service: On A Demand Collaboration Software There is an online learning and training (OLAT) provided on the HFU infrastructure. The OLAT allows for easy online course system supported by the versions of the system subversion (SVN) and the jabber instant messenger server based on XMPP. The OLAT allows users to create custom environment by enabling users to create working groups and adding relevant functionalities like chat and forums. Based on the high demand for OLAT and that it takes a whole day for installation and configuration, there was a need to have the CollabSoft. The CollabSoft is an OLAT system that is available on-demand and it is installed in a custom VM. The CollabSoft VM image comprises an apache server, the versioning system subversion, a database server, the Tomcat servlet container with the OLAT and an instant messaging server.

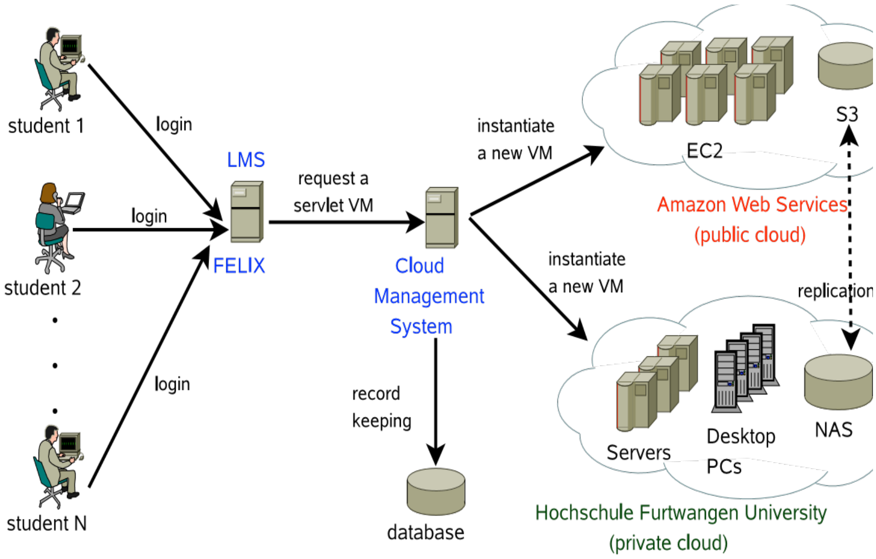

Figure 6: High level overview of running the servlet container application on the cloud [14]

6 Discussion

The HFU cloud E-Learning architecture is a typical example of advancement in technology and successful implementation of cloud E-Learning. This is typical of most tertiary institutions and educational establishments in advanced countries. The infrastructure, resources and skilled manpower available in these advanced tools makes this possible. Cloud E-Learning lowers the cost related to traditional E-Learning in terms of classroom and educational materials making E-Learning cheaper for developing nations. There are no infrastructural requirements with the attendant cost implications, hence all that is needed is a PC or mobile phone to connect and access E-Learning applications on the cloud. E-Learning has the capacity to improve education, training, knowledge, skills and creativity in developing nations thereby boosting economic growth and national development. Cloud E-Learning also serves as a catalyst to enhance technological development.

On the other hand, utilizing cloud E-Learning in developing countries especially African countries is an uphill task because of lack of cloud infrastructure needed to acquire latest hardware and software required for E-Learning. Poor African countries lack IT infrastructure, but are able to use mobile phones with an optimum access to data services. Cloud E-Learning methods applied in industrialized countries cannot be fully adopted by developing countries. Most African countries do not have the primary infrastructure required to enjoy the benefits of a cloud-based E-Learning system. There are various reasons for this. The first is the erratic power supply being experienced in these countries including Nigeria. Although mobile phones can be used, the issue of battery time makes this option unrealistic. Secondly, the bandwidth is not cheap and coverage is not adequate. In addition, Internet access is not of good quality and even when it is available, it is not accessible due to power outages. The most critical of these issues is that the continent is still lacking cloud service providers that will provide the highly needed infrastructure for the cloud based E-Learning. However, most Internet service providers on the continent are improving their services and a lot of changes are noticeable. Despite this slight optimism, cloud computing is generally still in its infancy in Africa. In addition, cloud E-Learning architecture like the HFU E-Learning cloud is not available in most tertiary institutions in Africa.

7 Conclusion

Cloud computing provides on-demand, scalable, and elastic services to users over the internet. Cloud computing offers SaaS, PaaS and IaaS services using the private, public, community or hybrid cloud. E-Learning involves the utilization of hardware and software provided through the Internet to users for the purpose of improving education. E-Learning has a unique architectural layer such as application, management, software and others. A typical application of cloud-based E-Learning in a university environment adopted by HFU has been discussed. Cloud computing is definitely enhancing the way E-Learning is being conducted. Most E-Learning infrastructures in Africa are not cloud-based. It is recommended that further studies are conducted in this area.

References

[1] F. F. Ahmed. “Comparative Analysis for Cloud Based E-Learning”. In: *Procedia Computer Science* 65 (2015). International Conference on Communications, management, and Information technology (ICCMIT’2015), pages 368–376. ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2015.09.098.

[2] I. Arpaci. “Antecedents and Consequences of Cloud Computing Adoption in Education to Achieve Knowledge Management”. In: *Computers in Human Behavior* 70 (2017), pages 382–390. ISSN: 0747-5632. DOI: https://doi.org/10.1016/j.chb.2017.01.024.

[3] U. J. Bora and M. Ahmed. “E-learning using Cloud Computing”. In: *International Journal of Science and Modern Engineering* 1.2 (2013), pages 9–12.

[4] S. Cerón-Figueroa, I. López-Yáñez, W. Alhalabi, O. Camacho-Nieto, Y. Villuendas-Rey, M. Aldape-Pérez, and C. Yáñez-Márquez. “Instance-based Ontology Matching for E-Learning Material using an Associative Pattern Classifier”. In: *Computers in Human Behavior* 69 (2017), pages 218–225. ISSN: 0747-5632. DOI: https://doi.org/10.1016/j.chb.2016.12.039.

[5] A. Dukhanov, M. Karpova, and K. Bochenina. “Design Virtual Learning Labs for Courses in Computational Science with Use of Cloud Computing

Technologies". In: *Procedia Computer Science* 29 (2014). 2014 International Conference on Computational Science, pages 2472–2482. ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2014.05.231.

[6] A. F. Hegazy, A. E. Khedr, and Y. A. Geddawy. "An Adaptive Framework for Applying Cloud Computing in Virtual Learning Environment at Education a Case Study of "AASTMT"". In: *Procedia Computer Science* 65 (2015). International Conference on Communications, management, and Information technology (ICCMIT'2015), pages 450–458. ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2015.09.121.

[7] M. S. Jamwal and C. Jamwal. "Cloud Computing for E-learning". In: *Advances in Computer Science and Information Technology* 2.8 (2015), pages 26–29.

[8] M. A. H. Masud and X. Huang. "A Novel Approach for Adopting Cloud-Based E-learning System". In: *2012 IEEE/ACIS 11th International Conference on Computer and Information Science*. May 2012, pages 37–42. DOI: 10.1109/ICIS.2012.10.

[9] M. A. H. Masud and X. Huang. "An E-learning System Architecture based on Cloud Computing". In: *International Journal of Computer, Electrical, Automation, Control and Information Engineering* 6.2 (2012), pages 255–259.

[10] P. M. Mell and T. Grance. *SP 800-145. The NIST Definition of Cloud Computing*. Technical report. Gaithersburg, MD, United States, 2011.

[11] P. Pocatilu, F. Alecu, and M. Vetrici. "Measuring the Efficiency of Cloud Computing for E-Learning Systems". In: *W. Trans. on Comp.* 9.1 (Jan. 2010), pages 42–51. ISSN: 1109-2750.

[12] Ş. A. Rădulescu. "A Perspective on E-learning and Cloud Computing". In: *Procedia - Social and Behavioral Sciences* 141 (2014). 4th World Conference on Learning Teaching and Educational Leadership (WCLTA-2013), pages 1084–1088. ISSN: 1877-0428. DOI: https://doi.org/10.1016/j.sbspro.2014.05.182.

[13] G. Riahi. "E-learning Systems Based on Cloud Computing: A Review". In: *Procedia Computer Science* 62 (2015). Proceedings of the 2015 International Conference on Soft Computing and Software Engineering (SCSE'15), pages 352–359. ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2015.08.415.

[14] A. Sulistio, C. Reich, and F. Doelitzscher. "Cloud Infrastructure & Applications – CloudIA". In: *Cloud Computing: First International Conference, CloudCom 2009, Beijing, China, December 1-4, 2009. Proceedings*. Edited by M. G. Jaatun, G. Zhao, and C. Rong. Berlin, Heidelberg: Springer, 2009, pages 583–588. ISBN: 978-3-642-10665-1. DOI: 10.1007/978-3-642-10665-1_56.

Performance Modelling for Multiple Virtual Machines Live Migration

VMware vMotion based Study

Mohamed Esam Elsaid[a], Ahmed Shawish[b,c], and Christoph Meinel[a]

[a]Hasso Plattner Institute for Digital Engineering, University of Potsdam, Germany
[b]The Open Arab University, Kuwait
[c]Ain Shams University, Egypt
firstname.lastname@hpi.uni-potsdam.de

Live migration of virtual machines is one of the essential features in cloud computing environments. Servers load balance, power saving and dynamic resource management are all dependent on the live migration feature. However, live migration has an overhead on the servers CPU, memory, network throughput, and power consumption. So it is worth to study the parameters that control this overhead and how much they impact it. Most of the current published analysis focus on the open source Xen or KVM hypervisor with less attention to the widely used VMware. Even thought, many studies focus on a single virtual machine performance aspect at a time and the others propose new migration techniques for low latency migration. In this paper, we provide a comprehensive performance analysis for live migration with VMware. Using a VMware test-bed, three different types of applications with diverse characteristics have been adopted to analyze the live migration of single and concurrent multiple virtual machines. The worst and the best scenarios have been examined to study all possible aspects related to this vital function in the cloud data-centers. The extensive analysis led us to several empirical models that can be used for live migration cost estimation and providing overhead aware resources management techniques.

1 Introduction

Cloud computing moves the data processing and storage away from desktop and portable PCs into large powerful data-centers to gain more reliability and reduce the cost. This would never have happened without the virtualization. In fact, virtualization features such as flexible resource provisioning, isolation, cloning, resource sharing and data migration have significantly improved the reliability and utilization of the IT resources. Infrastructure, Platform and Software as a service (IaaS, PaaS and SaaS), are offered using an illusion of availability in resources as demanded by the Cloud users to rapidly satisfy their requirements [2]. This virtual availability of resources utilizes the hardware efficiently, facilitates rapid scaling, while saving power and cost. This is quite different from the earlier infrastructure models where the enterprises had to invest huge cost in building the IT infrastructure resource including cooling and IT staff.

Live Migration is a powerful feature in cloud computing and virtual data-centers. It allows the running online Virtual Machines (VM) to be moved from a physical server to another with very little interruption which allows seamless movement of online servers in LAN or in MAN scale without asking clients to disconnect and reconnect [19].

Live migration is supported by VMware (vMotion), Xen (XenMotion), Microsoft Hyper-V and Redhat KVM [16]. Resource management in virtual data-centers as well as servers load balancing, online maintenance, fault tolerance and power saving are all dependent on VMs live migration feature. On the other hand, it is important to note that live migration has a cost that affect the data-centers' physical resources like memory, network and power consumption.

Most of the current analysis [2] [19] [16] [3] [1] [17] [22] [20] is focussing on the open source Xen with far less attention to the widely used VMware, where most of these analysis are theoretical and focused on a single performance aspect at a time. Other researches [5] [18] [9] [6] [23] [4] are focussing on proposing new techniques for low latency migration. In general, a full analysis of this vital function covering all of its overhead aspects is crucially needed to avoid random VMs migration that can significantly affect the efficiency of resources utilization.

This paper provides a deep analysis of the live migration cost and proposes empirical models for live migration time, memory and power consumption. Using a real testing lab, the proposed models are verified by different running applications, different VM memory size and different number of VMs migration in parallel. We also explain in this paper the theoretical background of each proposed model.

The proposed migration overhead models can be used for giving an estimation to data-center admins about the expected cost of doing live migration for certain VM based on its configuration and workload. This feature is very useful to avoid blind live migration that may result in bottlenecks in physical resources and degrade the performance of a running application. The proposed models are also useful for power saving and load balance research study in virtual environments that rely on live migration modeling. In this domain, live migration cost aware resource management algorithms can be proposed.

The remainder of this paper is organized as follows: Section 2 discusses the background about live migration overhead and the related work. Section 3 shows the details about obtaining the performance models and Section 4 presents the testbed configuration and the resultant models with considering the theoretical verification beyond the resulted graphs. The paper is then concluded in Section 5.

2 Background

This section, firstly, provides a comprehensive overview on the live migration cost. It then reviews the related work and discusses their limitations.

2.1 Live Migration Cost

Live migration cost is classified into energy and performance overhead on the running machines as well as the live migration execution cost [19]. Execution costs are the total migration time and migration down time. However, the physical machines overhead are the increase in CPU utilization, network throughput, and power consumption [19]. There are two types of live migration; post-copy and pre-copy technique [9]. Pre-copy is the less interruptive technique in machine migration and therefore commonly used in industry which is why we will focus on it in this paper.

Live migration in iterative pre-copy technique is used in Xen and VMware. The technique has six major phases, namely:

1. Initialization: Initiating the migration by selecting the VM to be migrated and selecting the target machine.
2. Reservation: The source machine sends a request to the target machine for resources reservation and the target machine answers with an acknowledgment after reserving the required resources for the migration.
3. Iterative pre-copy: The entire RAM is sent in the first iteration, and then pages modified during the previous iteration are transferred to the destination. Using shadow page table for memory dirty pages mapping.
4. Stop-and-Copy: When the stop conditions are met, the VM is halted on the source for a final transfer round. The stop conditions in the Xen platform are:

   - Less than 50 pages are dirtied during the last pre-copy iteration.
   - Or 29 pre-copy iterations have been carried out.
   - Or more than 3 times the total amount of RAM allocated to the VM has been copied to the destination.

   While the stop conditions for VMware are [15]:

   - Less than 16 megabytes of modified pages are left.
   - Or there is a reduction in changed pages of less than 1 megabyte.

   At the same round of stop-and-copy while transferring the final dirty pages, the migrated VM's CPU state is transferred to the destination.
5. Commitment: the destination host checks if it has successfully received a consistent copy of the migrated VM. Then the target machine sends a message telling the source that it has successfully synchronized the migrated VM states.

6. Activation: After target host informs source host that it has synchronized their states, source VM can be discarded. The migrated VM running on target host is the primary host now and takes over the services offered by source VM.

2.2 Related Work

Migration time and downtime analysis are studied in [2] [19] [16] [3] [1] [17] [22] [20]. Live Migration time and downtime are mathematically modelled in [2] [19] [3] [1]. A very useful comparison between live migration in VMware, Xen, KVM and Hyper-V is done in [16] in terms of the migration time, downtime and migration volume; in this paper we focus only on VMware hypervisor as a commonly used hypervisor especially in enterprise environments. We do not study the live migration time only, but we study the other migration cost factors like memory, network and power consumption overhead. In [17], RAM intensive, CPU intensive and Disk IO intensive benchmarks are used to measure the migration time with different CPU capacities. However, there are no models proposed in [16] and [17] to estimate live migration time and impact on CPU, network and network utilization, and this is what we study in the paper to support the live migration action decision with overhead predictive models.

Different novel live migration algorithms are proposed in [5] [18] [9] to minimize live migration cost for single VM. In [6], the authors address Inter-Rack Live Migration (IRLM) to minimize VMs migration cost between different physical servers racks. IRLM depends on minimizing the migration volumes by reducing the amount of memory content to be transferred by using deduplication technique. In this paper, we do not focus on doing enhancements on the live migration algorithm of VMware vMotion, our objective is to study the existing algorithm overhead. Resource reservation technique is applied in [23] for multiple VMs parallel migration. This resource reservation technique performance is compared for different benchmarks using Xen environment in terms of migration time and downtime. However, there are no models proposed from these results, also other hypervisors like VMware or Hyper-V are not considered in this study. Authors in [4] propose an analytical Markovian model for inter-data-centers network capacity planning in order to enhance multiple VMs parallel migration performance in a federate cloud.

To the best of our research knowledge, most of the research papers address the Xen hypervisor as an open source, while less attention has been given to the widely used VMware hypervisor. Moreover, none of the previous work gave attention to different applications running on multiple VMs live migration modeling.

In this paper, we continue the research on the proposed live migration cost modelling in [7] and [8]. In [7], we have presented the live migration impact on virtual data-center performance for a single VM. In [8], empirical modelling for multiple virtual machines is proposed using an extensive memory usage application.

This paper provides a deeper analysis of the existing live migration algorithm in VMware vMotion without any modification in the algorithm details. It focuses on VMware as a commonly used enterprise hypervisor in modern data-centers. In addition, it discusses the theoretical background beyond the obtained empirical models.

3 Performance Modelling

To obtain empirical models of live migration overhead, we used the following methodology. Firstly, we built a VMware testbed; with configuration details as in the next section. Then we run different live migration sessions for different numbers of parallel VMs and different memory size, because memory size is the dominant factor in live migration cost. The running application is a main player in live migration overhead amount.

In this study, we considered three applications; Linpack benchmark [13], network stress and idle VM. Linpack benchmark is a CPU and memory intensive benchmark which is the worst case scenario for an application from resource utilization perspective. Idle VM is the best case scenario for a VM utilization and finally network stress which represents virtual machines in cloud computing environment. This is because the VMs in the cloud environment are mainly accessed by remote users through the network and so the applications input/outputs should be transmitted through the network. By these three applications, we validate the obtained reading of live migration overhead performance metrics.

For each live migration session, we use VMware vCenter server performance tool to measure the migration time in seconds, the active memory utilization change in kB, the network throughput in kBps change, and the power consumption increase in Watt. From the obtained results, we get several distracted points that can be fitted to a well-defined mathematical relation. To find the best fit relation, our target is to minimize the error between the obtained testing readings and the fitted curve line. This is done using MATLAB to achieve the objective function of minimum error in equation:

$$\begin{aligned} \text{Min.} \qquad & E_{\theta 1,\theta 2} = \frac{1}{m}\sum_{i=1}^{m} h_\theta(x^{(i)} - y^{(i)})^2 \\ & h_\theta(x^{(i)}) = \theta_1 + \theta_2 \end{aligned} \tag{1}$$

Where:
i: test reading number
x: is the horizontal scale variable
$E_{\theta 1,\theta 2}$: Square Error as function of θ_1 and θ_1
$y^{(i)}$: testing reading value
m: Total number of readings
In the next section, we discuss the testbed configuration details, experiment results, performance-modelling and the theoretical validation of the obtained models.

4 Results Analysis and Modelling

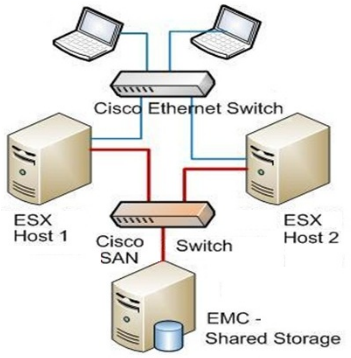

Figure 1: Testing Lab Network Diagram

To study live migration performance modeling, we have built a VMware lab setup with the following specifications: 2 physical hosts (Dell PowerEdge 2950). Each host has 8 CPU x 2.992 GHz Intel(R) Xeon, 20GB RAM, 2 NICs and 1 HBA with 2 Fiber ports/card. VMware vSphere 5.1 hypervisor is used vCenter server appliance for live migration and performance monitoring.

As shown in Figure 1; both hosts are connected to a shared EMC2 VNX block storage [12] with 1TB LUN via FC-SAN. The SAN Switch is Cisco with 4Gbps ports. The Ethernet switch is Cisco with 1Gbps ports. Live migration process utilizes the Ethernet switch [10]. The two physical hosts are configured as a cluster that is managed by the VMware vCenter Server which manages the cluster resources and includes the vMotion feature [14]. Performance metrics are also gathered using the vCenter Server performance monitoring interface. The VMs that are used in this migration are Linux Ubuntu 12.04 (32bit) with 4 vCPU. The testing benchmark is Linpack, network stress; Apache Bench (AB) and idle VM. Linpack is a CPU and RAM intensive benchmark [13]; which is the worst case scenario for a running application. The network stress application that we have used is Apache Bench (AB). Apache Bench tool stresses the web servers with lots of requests to test the servers' response; which is also a stress on the network. Idle VM is simply an idle Ubuntu OS VM; with no running applications. So the workload comes only from the Ubuntu OS events.

The VM RAM size is the most effective parameter in the migration performance [19]. So we test the impact of live migration on the datacenter performance with different memory sizes to have different migration volumes. The VM RAM size varies between 1GB, 2GB, 4GB and 8GB. The VM is powered on, the benchmark is run and after 5 minutes at least, the VM migration is started from one of the physical hosts to the other in order to distinguish between the benchmark impact and the migration impact on performance. After the migration is finished, the migration time is calculated and the impact on the target host performance is monitored. Finally the benchmark is stopped.

Based on the testbed configuration in section 4, we have run several testing scenarios in order to obtain empirical models for live migration performance that can be used for live migration cost prediction. Using the above infrastructure the testing sequence was as following; the migration was done 5 times for each memory configuration value of the same number of VMs running an application. For example in Linpack application, we run 1VM with 1GB RAM 5 times, 5 times for 2 GB RAM VMs, 5 times for 4GB RAM and 5 times for 8 GB RAM. The same steps are repeated for a different number of VMs at the same time and for different applications.

4.1 Migration Time Modelling

4.1.1 Observations

Migration time is the period between the VM migration request initialization and having the VM activated at the destination server. In this experiment, we follow VMware vsphere client tasks list in order to know the VM migration start and end times. The difference between the two times is the VM migration time. Figures 2, 3 and 4 show the migration time testing results versus the active memory size over the network throughput due to migration. The results show the migration time duration for different number of VMs, with different memory sizes and using Linpack, network intensive and idle VM applications. In Figure 2, the migration time of Linpack benchmark is presented and from the obtained results, we see the following observations:

- With increasing the number of VMs in parallel migration, the migration time increases; so 1 VM migration has the least migration time and 4 VMs migration has the longest migration time.

- The starting point of the line in the vertical axis has the lowest value for 1 VM migration and has the highest value for 4 VMs migration.

- The regression of the resultant testing points shows linear curve fitting; as represented in equation (1) which assures the proposed models in [7] and [8].

  $$T_{mig} = a.(\frac{V_{mem}}{R_s}) + b \tag{2}$$

  R_s: is the source host network throughput increase, V_{mem} is the source host active memory size before migration starts, T_{mig} is the migration duration time and a and b are constants that change with the datacenter hardware configurations.

- The linear relation slope increases with having more number of parallel VMs due to the greater impact on the migration time with having more VMs at the same value of the memory size over the rate.

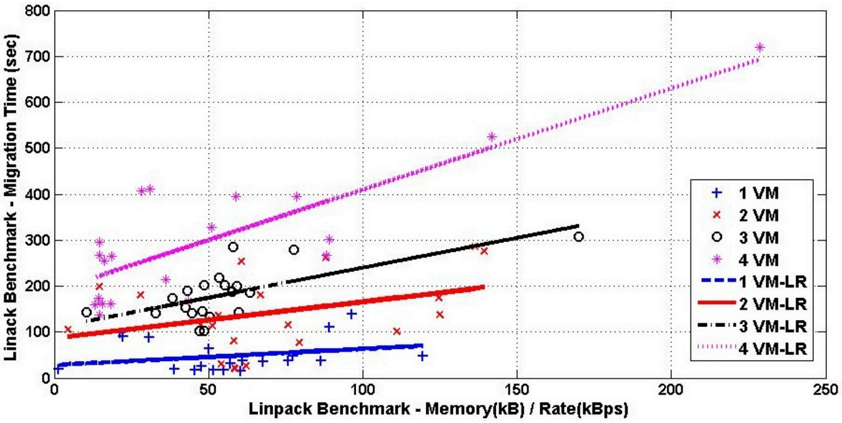

Figure 2: Avg. Migration Time – Linpack Benchmark

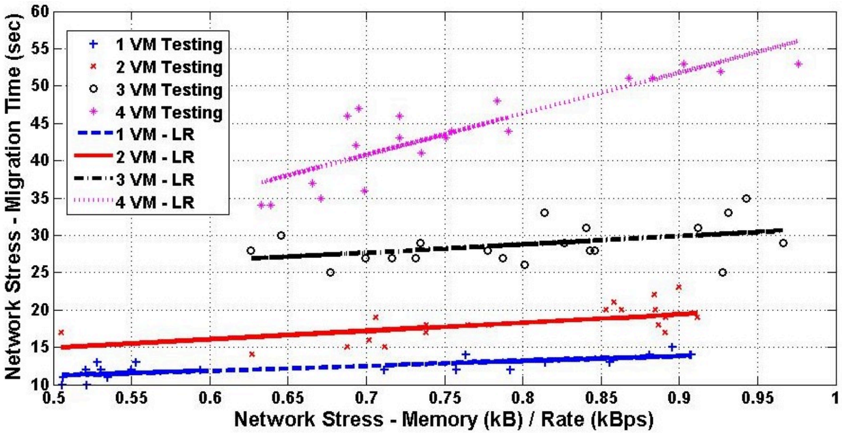

Figure 3: Avg. Migration Time – Network Stress Application

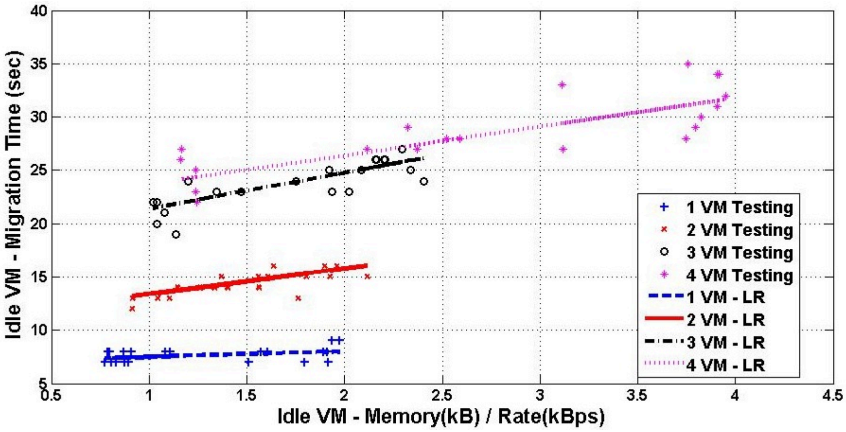

Figure 4: Avg. Migration Time – Idle VM

4.1.2 Theoretical Background – Migration Time

The theoretical background of the migration time linear modelling in Figures 6, 7 and 8 can be explained as following. Equation set (3) represents the migration time analytical modelling proposed in [17].

$$
\begin{aligned}
T_{mig} &= \sum_{i=0}^{n} T_i = \frac{V_{mem}}{R_s} \cdot \frac{1-\lambda^{n+1}}{1-\lambda} \\
\lambda &= \frac{D}{R_s} \\
T_{Norm} &= \frac{T_{mig}}{\frac{V_{mem}}{R_s}} = \frac{1-\lambda^{n+1}}{1-\lambda}
\end{aligned}
\qquad (3)
$$

D is the dirty pages rate in kBps, n is the number of live migration iterations and R_s is the transmission rate in kBps. For successful live migration λ should be less than 1; otherwise the migration copy iterations will exceed the stopping condition timeout and fail. Figure 5 shows the relationship in the set of equations (3) with different values of n and λ. The normalized migration time can be fitted to a linear relation especially for small values of (n) or (λ). With higher values of (n) or (λ), the linear fitting error increases.

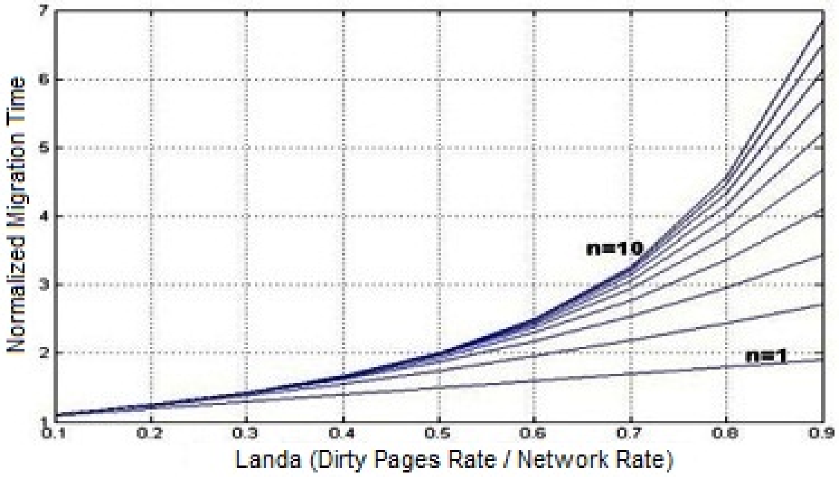

Figure 5: Migration time relation between λ and n

4.2 Network Throughput Modelling

4.2.1 Observations

Network average rate is the average throughput at which data was transmitted from the physical host NIC card during the migration time interval. This represents the consumed bandwidth of the network in Bps for live migration process.

Figures 6, 7 and 8 show the relation between the network rate and active memory size of the source host. From these figures, we highlight the following observations:

- The relation between the network rate and the active memory size for the three curves can be modelled as an exponential relation.

  $$R_s = \alpha e^{-V_{Mem}} + \beta \tag{4}$$

  Where:
  R_s: Source host throughput increase
  V_{Mem}: Source host active memory size before migration α and β are constants

- Values of the x and y axes increase with the number of parallel VMs to be migrated.

- 1 VM curve has the lowest start point in x and y axis in the three figures because it has the lowest active memory size and rate.

- 4 VM curve has the highest network throughput in the three figures due to the largest active memory consumption.

- The curves may overlap at some points which is due to the change in the memory size with moving the x axis. So for example, 1 VM with 8GB memory will consume more throughput than 2 VMs with 2GB memory size.

4.2.2 Theoretical Background – Network Rate

Live migration process is managed by the VMware cluster vCenter server which uses the Transmission Control Protocol / Internet Protocol (TCP / IP) in the network layers 3 and 4 for the live migration management and the iterative copies of memory pages. As mentioned in [11] and illustrated in Figure 9, TCP congestion window is initially set to a maximum segment size. The sender then sends a burst of this

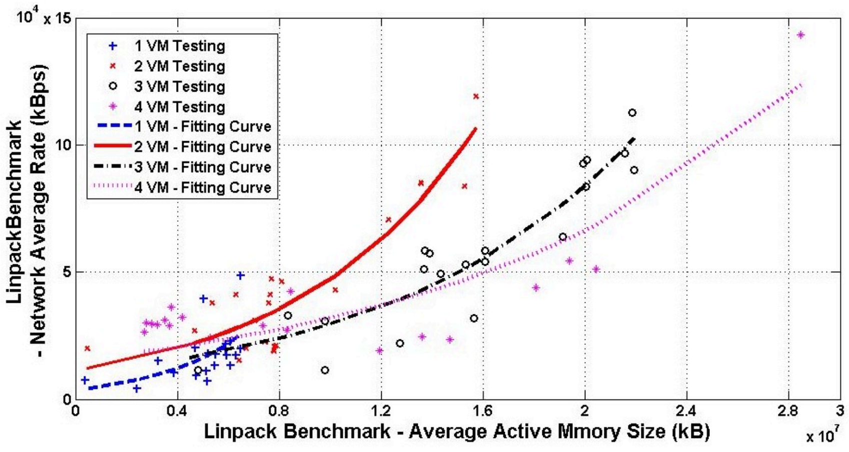

Figure 6: Avg. Network Rate – Linpack Benchmark

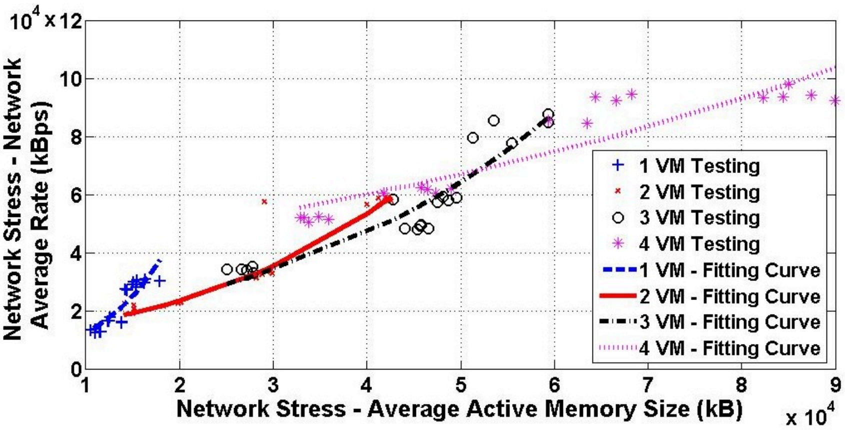

Figure 7: Avg. Network Rate – Network Stress Application

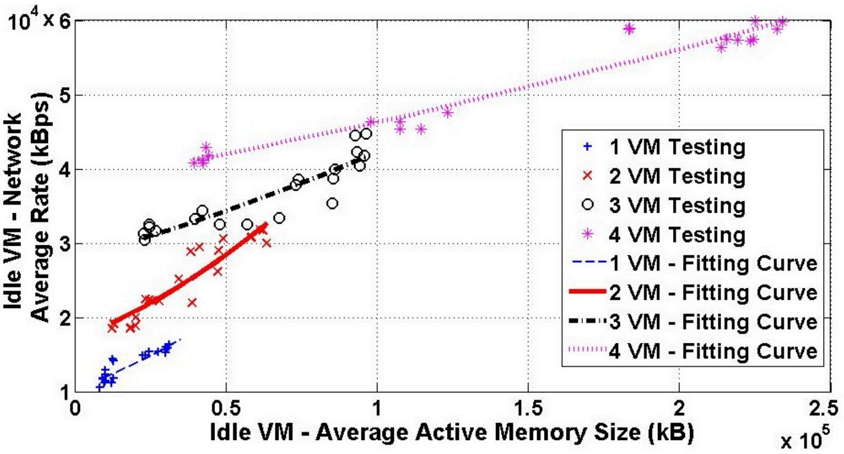

Figure 8: Avg. Network Rate – Idle VM

number of bytes. If the burst is received (timer does not expire) then it doubles the congestion window and sends a burst of 2 * max segment size. It continues increasing the congestion window until a burst is lost (time-out). Now the sender knows how much the network can handle. This is called slow start. The slow start phase has almost an exponential relationship between the segment size and the number of rounds. When a timeout happens the threshold is set to one half the current congestion window, and increases in a saw-tooth shape [21].

Because live migration is running for a short time, few minutes on average, we can say that it is mainly controlled by the slow start phase in TCP which has an exponential relationship between the number of round trips and the number of segments to be sent in each round trip. This represents also the relation between the active memory size and the network rate. With the x-axis indicating an increased amount of active memory content, resulting in a higher number of round trips, and

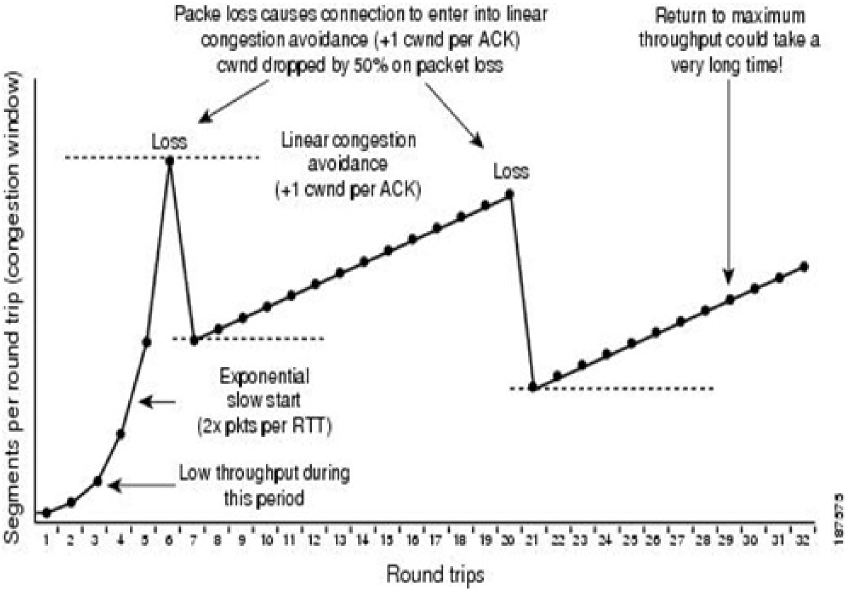

Figure 9: TCP congestion control [11]

with the y-axis indicating a larger number of segments that need to be sent at the same time, together the overall network throughput is high.

4.3 Power Consumption Modelling

4.3.1 Observations

Power consumption is always a critical point in datacenters operations due to the AC power high running cost and impact on CO_2 emissions. One of the main benefits in resource virtualization is optimizing the power usage for IT systems. So we study in this sub-section, the power consumption overhead due to VMs live migration. In this experiment, we use VMware vCenter server performance tool to measure the peak increase in the power consumption in Watt of each physical host after each migration test and map this change with the VM memory size and number of VMs; as shown in Figures 10, 11 and 12. From the charts in the below power consumption figures, we note the following:

- Power consumption peak increase has linear relation with the transmission rate. This linear relation between the power consumption and the transmission rate is also obtained in [17], but for Xen environment. In this paper, we prove the same relation also for VMware environment.

$$P_{mig} = \frac{dE_{mig}}{dt} = c\,\frac{dV_{mig}}{dt} = c\,R \tag{5}$$

  Where:
  P_{mig}: Peak power increase after live migration
  E_{mig}: Peak energy increase after live migration

V_{mig}: Migration volume
R: Migration rate

- Linpack application shows the highest peak power increase due to high network rate, in contrary the idle VM shows the lowest peak power consumption.
- The graph lines overlap at some points because high memory size for low number of VMs may consume higher power than low memory size for more number of VMs; for example in Figure 11, 3 VMs with 8GB memory consume more power than 4 VMs with 4GB memory.

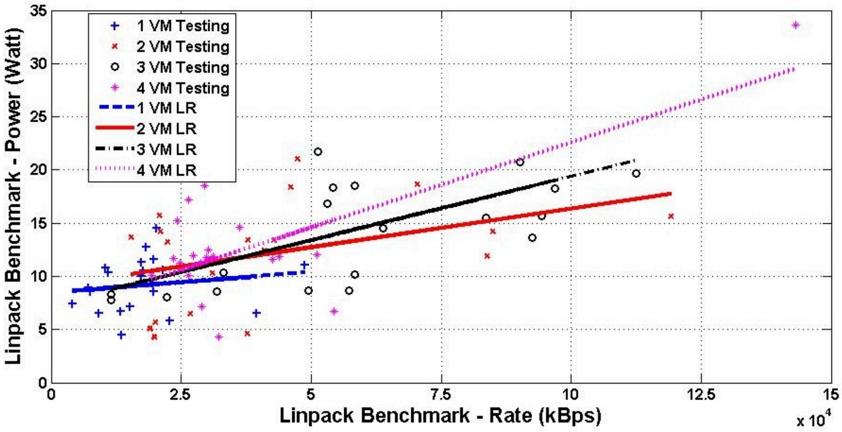

Figure 10: Power consumption – Linpack Benchmark

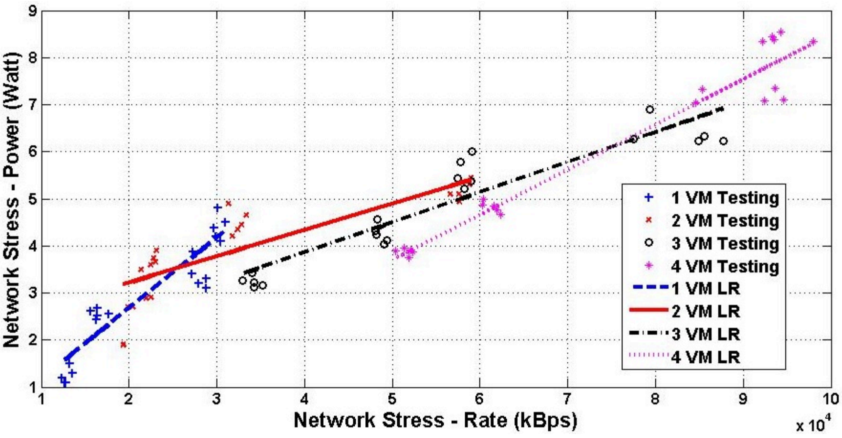

Figure 11: Power consumption – Network Stress Application

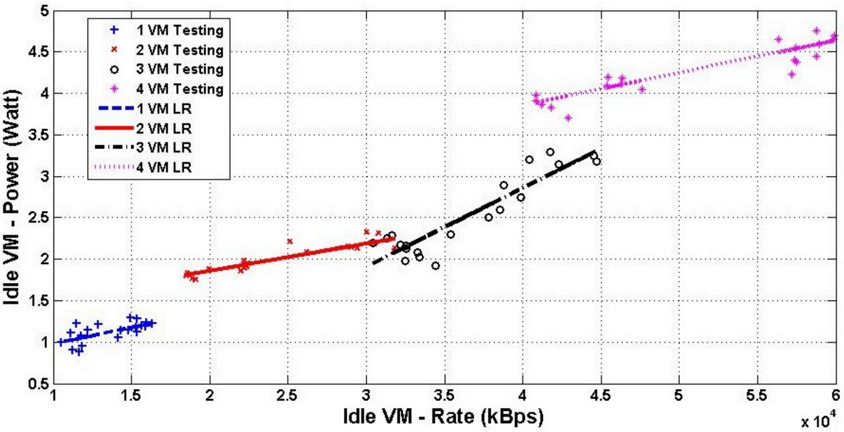

Figure 12: Power consumption – Idle VM

4.3.2 Theoretical Background – Power Consumption

Migration rate is the dominant factor for power consumption during the migration process; with higher network rate the migration duration becomes shorter and the power consumption increases [17]. Based on the proposed modelling for live migration energy consumption in [17], there is a direct relation between the consumed energy E_{mig} and the migration volume V_{mig}; equation (6) which represents also the direct relation between the consumed power and the migration rate:

$$E_{mig} = \alpha\ V_{mig} + \beta \tag{6}$$

5 Conclusion

Live migration is an essential feature in virtual datacentres and cloud computing environments. Resource management, power saving and servers load balance rely on live migration process. In this paper, we proposed empirical models for live migration overhead in a VMware cluster with single and multiple VMs migration using three different applications that utilize different VM memory sizes. The results show that from the VM active memory size and using the proposed empirical models, live migration time, network throughput and power consumption overhead can be obtained as the main VM migration cost parameters. The proposed models are theoretically verified to explain the obtained relations meaning. These empirical models can be used for live migration overhead prediction and resource management optimization in load balance and power saving as well as avoiding bottlenecks during live migration process. This prediction can be achieved using machine learning algorithms and using the obtained empirical models. When the live migration cost can be predicted, this means that the live migration timing or the number of migrated VMs can be optimized to avoid any resources bottlenecks during the live migration process.

References

[1] S. Akoush, R. Sohan, A. Rice, A. W. Moore, and A. Hopper. "Predicting the Performance of Virtual Machine Migration". In: *2010 IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*. Aug. 2010, pages 37–46. DOI: 10.1109/MASCOTS.2010.13.

[2] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia. "A View of Cloud Computing". In: *Commun. ACM* 53.4 (Apr. 2010), pages 50–58. ISSN: 0001-0782. DOI: 10.1145/1721654.1721672.

[3] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I. Stoica, et al. *Above the clouds: A Berkeley View of Cloud Computing*. Technical report. Technical Report UCB/EECS-2009-28, EECS Department, University of California, Berkeley, 2009.

[4] W. Cerroni. "Multiple Virtual Machine Live Migration in Federated Cloud Systems". In: *2014 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. Apr. 2014, pages 25–30. DOI: 10.1109/INFCOMW.2014.6849163.

[5] C. Clark, K. Fraser, S. Hand, J. G. Hansen, E. Jul, C. Limpach, I. Pratt, and A. Warfield. "Live Migration of Virtual Machines". In: *Proceedings of the 2Nd Conference on Symposium on Networked Systems Design & Implementation - Volume 2*. NSDI'05. Berkeley, CA, USA: USENIX Association, 2005, pages 273–286.

[6] U. Deshpande, U. Kulkarni, and K. Gopalan. "Inter-rack Live Migration of Multiple Virtual Machines". In: *Proceedings of the 6th International Workshop on Virtualization Technologies in Distributed Computing Date*. VTDC '12. Delft, The Netherlands: ACM, 2012, pages 19–26. ISBN: 978-1-4503-1344-5. DOI: 10.1145/2287056.2287062.

[7] M. E. Elsaid and C. Meinel. "Live Migration Impact on Virtual Datacenter Performance: Vmware vMotion Based Study". In: *2014 International Conference on Future Internet of Things and Cloud*. Aug. 2014, pages 216–221. DOI: 10.1109/FiCloud.2014.42.

[8] M. E. Elsaid and C. Meinel. "Multiple Virtual Machines Live Migration Performance Modelling – VMware vMotion Based Study". In: *2016 IEEE International Conference on Cloud Engineering (IC2E)*. Apr. 2016, pages 212–213. DOI: 10.1109/IC2E.2016.9.

[9] M. R. Hines, U. Deshpande, and K. Gopalan. "Post-copy Live Migration of Virtual Machines". In: *SIGOPS Oper. Syst. Rev.* 43.3 (July 2009), pages 14–26. ISSN: 0163-5980. DOI: 10.1145/1618525.1618528.

[10] *http://pubs.vmware.com/*.

[11] *http://www.cisco.com/c/en/us/td/docs/nsite/enterprise/wan/wan_optimization/wan_opt_sg/chap06.html*.

[12] *http://www.emc.com/*.

[13] *http://www.netlib.org/linpack/*.

[14] *http://www.vmware.com/products/vcenter-server/*.

[15] B. Hu, Z. Lei, Y. Lei, D. Xu, and J. Li. "A Time-Series Based Precopy Approach for Live Migration of Virtual Machines". In: *2011 IEEE 17th International Conference on Parallel and Distributed Systems*. Dec. 2011, pages 947–952. DOI: 10.1109/ICPADS.2011.19.

[16] W. Hu, A. Hicks, L. Zhang, E. M. Dow, V. Soni, H. Jiang, R. Bull, and J. N. Matthews. "A Quantitative Study of Virtual Machine Live Migration". In: *Proceedings of the 2013 ACM Cloud and Autonomic Computing Conference*. CAC '13. Miami, Florida, USA: ACM, 2013, 11:1–11:10. ISBN: 978-1-4503-2172-3. DOI: 10.1145/2494621.2494622.

[17] H. Liu, H. Jin, C.-Z. Xu, and X. Liao. "Performance and Energy Modeling for Live Migration of Virtual Machines". In: *Cluster Computing* 16.2 (June 1, 2013), pages 249–264. ISSN: 1573-7543. DOI: 10.1007/s10586-011-0194-3.

[18] C. Sagana, M. Geetha, and R. C. Suganthe. "Performance Enhancement in Live Migration for Cloud Computing Environments". In: *2013 International Conference on Information Communication and Embedded Systems (ICICES)*. Feb. 2013, pages 361–366. DOI: 10.1109/ICICES.2013.6508339.

[19] A. Strunk. "Costs of Virtual Machine Live Migration: A Survey". In: *2012 IEEE Eighth World Congress on Services*. June 2012, pages 323–329. DOI: 10.1109/SERVICES.2012.23.

[20] W. Voorsluys, J. Broberg, S. Venugopal, and R. Buyya. "Cost of Virtual Machine Live Migration in Clouds: A Performance Evaluation". In: *Cloud Computing: First International Conference, CloudCom 2009, Beijing, China, December 1-4, 2009. Proceedings*. Edited by M. G. Jaatun, G. Zhao, and C. Rong. Berlin, Heidelberg: Springer, 2009, pages 254–265. ISBN: 978-3-642-10665-1. DOI: 10.1007/978-3-642-10665-1_23.

[21] M. Welzl. *Network Congestion Control: Managing Internet Traffic*. John Wiley & Sons, 2005.

[22] Y. Wu and M. Zhao. "Performance Modeling of Virtual Machine Live Migration". In: *2011 IEEE 4th International Conference on Cloud Computing*. July 2011, pages 492–499. DOI: 10.1109/CLOUD.2011.109.

[23] K. Ye, X. Jiang, D. Huang, J. Chen, and B. Wang. "Live Migration of Multiple Virtual Machines with Resource Reservation in Cloud Computing Environments". In: *2011 IEEE 4th International Conference on Cloud Computing*. July 2011, pages 267–274. DOI: 10.1109/CLOUD.2011.69.

Aktuelle Technische Berichte des Hasso-Plattner-Instituts

Band	ISBN	Titel	Autoren / Redaktion
121	978-3-86956-430-2	**Towards version control in object-based systems**	Jakob Reschke, Marcel Taeumel, Tobias Pape, Fabio Niephaus, Robert Hirschfeld
120	978-3-86956-422-7	**Squimera : a live, Smalltalk-based IDE for dynamic programming languages**	Fabio Niephaus, Tim Felgentreff, Robert Hirschfeld
119	978-3-86956-406-7	**k-Inductive invariant Checking for Graph Transformation Systems**	Johannes Dyck, Holger Giese
118	978-3-86956-405-0	**Probabilistic timed graph transformation systems**	Maria Maximova, Holger Giese, Christian Krause
117	978-3-86956-401-2	**Proceedings of the Fourth HPI Cloud Symposium "Operating the Cloud" 2016**	Stefan Klauck, Fabian Maschler, Karsten Tausche
116	978-3-86956-397-8	**Die Cloud für Schulen in Deutschland : Konzept und Pilotierung der Schul-Cloud**	Jan Renz, Catrina Grella, Nils Karn, Christiane Hagedorn, Christoph Meinel
115	978-3-86956-396-1	**Symbolic model generation for graph properties**	Sven Schneider, Leen Lambers, Fernando Orejas
114	978-3-86956-395-4	**Management Digitaler Identitäten : aktueller Status und zukünftige Trends**	Christian Tietz, Chris Pelchen, Christoph Meinel, Maxim Schnjakin
113	978-3-86956-394-7	**Blockchain : Technologie, Funktionen, Einsatzbereiche**	Tatiana Gayvoronskaya, Christoph Meinel, Maxim Schnjakin
112	978-3-86956-391-6	**Automatic verification of behavior preservation at the transformation level for relational model transformation**	Johannes Dyck, Holger Giese, Leen Lambers
111	978-3-86956-390-9	**Proceedings of the 10th Ph.D. retreat of the HPI research school on service-oriented systems engineering**	Christoph Meinel, Hasso Plattner, Mathias Weske, Andreas Polze, Robert Hirschfeld, Felix Naumann, Holger Giese, Patrick Baudisch, Tobias Friedrich, Emmanuel Müller
110	978-3-86956-387-9	**Transmorphic : mapping direct manipulation to source code transformations**	Robin Schreiber, Robert Krahn, Daniel H. H. Ingalls, Robert Hirschfeld
109	978-3-86956-386-2	**Software-Fehlerinjektion**	Lena Feinbube, Daniel Richter, Sebastian Gerstenberg, Patrick Siegler, Angelo Haller, Andreas Polze

ISBN 978-3-86956-432-6
ISSN 1613-5652