JOHANNES KEPLER
UNIVERSITY LINZ | JKU

TNF

Faculty of Engineering
and Natural Sciences

Seminar in Cloud Computing
IaaS Systems

Submitted by:

Markus Hiesmair
Jrgen Ratzenbck
Christoph Stengerlein

Created at:

Institut für ...

Graded by:

...

Linz, October 28, 2015

Affidavit

I hereby declare that the following dissertation ”Seminar - IaaS Systems” has been written only by the undersigned and without any assistance from third parties.

Furthermore, I confirm that no sources have been used in the preparation of this thesis other than those indicated in the thesis itself.

Linz, on December 5, 2015

Markus Hiesmair

Acknowledgment

Summary

Summary ...

Abstract

Abstract ...

Contents

Abbreviations

IaaS Infrastructure as a Service

PaaS Platform as a Service

SaaS Software as a Service

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview on Cloud computing

Nowadays software isn't just installed on an arbitrary computer for a specific user who can fulfil his given requirements by solving a task with it. Quite the contrary is the case as the significance of software has increased dramatically throughout any kind of business sector. The demand on software products these days is immense and therefore also the complexity and variety has experienced a huge growth over the last decade. Many years ago the Internet built up the fundament of accessing and sharing information worldwide and today applications and services, relying on complex and huge software ecosystems, give people around the globe the opportunity to use them any time and anywhere they want to satisfy their needs. To make this work this obviously needs a lot of resources accessible in the global network.

Here the famous and hyped term "Cloud computing", which describes the process of moving application and services to the internet (due to the schematic metaphor also denotes as "cloud"), comes into play. [4] In such intensive businesses with rare resources as we have it nowadays people have to concentrate on their specific tasks to be as productive, competitive and flexible as possible. Cloud computing supports this by providing a pool of resources allowing for sharing and scalable deployment of services, as needed, from almost any location, and for which the customer can be billed based on actual usage. [4]

How these resources are provided and shared depends on the specific requirements and can vary. Due to the common patterns of usages some different cloud types describing the strategy have established over time. [4]

- **Private Cloud**: The sharing of resources stays in-house and a specific organization is responsible for operating and maintaining the cloud infrastructure.

- **Community Cloud**: Several organizations having a common interest operate and maintain the shared cloud infrastructure. For the participating organizations such a solution can be very cheap if they agree on the community model.

- **Public Cloud**: An organization renting the cloud infrastructure from a specific provider who is responsible for it. The infrastructure is publicly available on a commerical basis.

- **Hybrid Cloud**: This is a mixture of the other existing types which can be tailored based on the concrete requirements for optimizing productivity. This can be for instance used if some data should be necessarily kept in-house and the rest could be outsourced in a Public Cloud.

Over the years different service models depending on the type of the provided resource have been established. Bascially they can be divided into three different types organized in a cloud computing stack with increasing abstraction level bottom-up.

IaaS basically means providing a shared pool of compute, storage and networking resources to end-users on a self-service basis. [5] This should help the end-users avoiding additional costs by buying dedicated hardware and setting up the instances to run their applications. They can easily manage and control the systems, in terms of operating system, network connectivity and storage and applications running on these instances but do not have to care about controlling and maintaining the cloud infrastructure. [4]

PaaS as the name already indicates provides the whole platform ”out-of-the-box” to the end-user. This includes things like the operating system or network

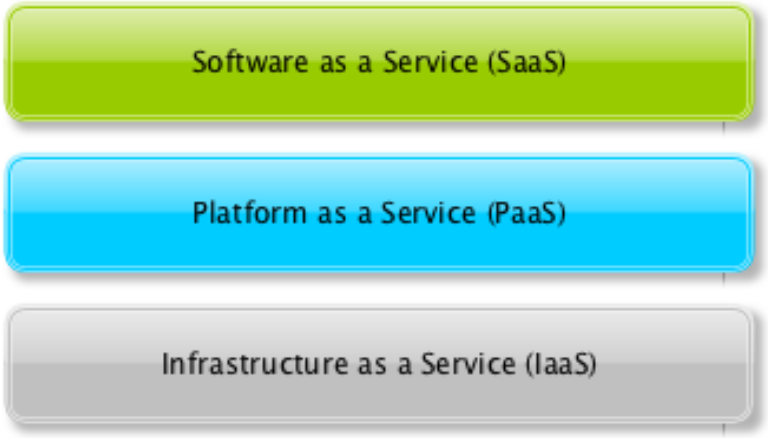

Figure 1.1: Stack of service models

connectivity which are completely managed by the provider. The user only has to deploy her applications to the cloud. [4]

SaaS abstracts the platform and infrastructure and serves the software living in the cloud as a usable service to the end-user. [4] This gives users instant access to such software without any special requirements such as downloading or installing and enables cross-platform as well as cross-device possibility.

Chapter 2

Advantages and Disadvantages of Cloud Computing Systems

The classical approach of deploying software, which was the case before the invention of cloud computing, is called ”Dedicated Hardware”. Using this approach companies buy hardware on their own which is dedicated only for the intended software, which should run on it. To provide a good level of service, usually companies buy hardware which can handle worst case scenarios and load peaks [6].

The problem of this approach is, that if the system runs on average work load, the full hardware capacities are not used and therefore resources are wasted (e.g. CPU cycles, storage space or RAM). If there are peaks in the work load this can result in a huge waste of resources or in a poor service of the software, if the hardware is not designed for such high peaks. Figure 2.1 shows some of these problems [6].

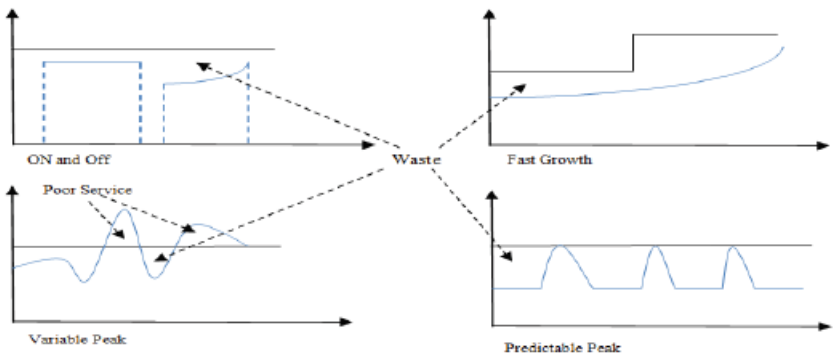

Figure 2.1: Dedicated Hardware Model - Utilization Waste [6]

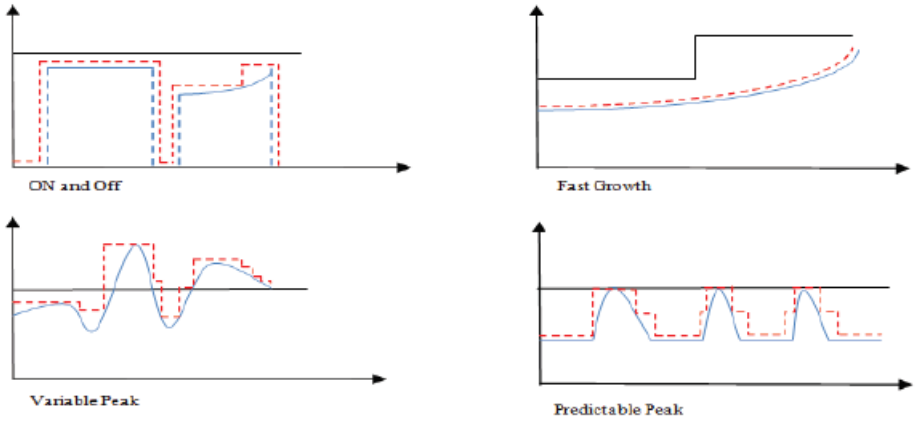

Figure 2.2: Cloud Computing Model - Dynamic Resources [6]

In cloud computing this waste of resources can be prevented by dynamically assigning resources when needed. For example, if the cloud computing system identifies that there is an overload, it can just add another virtual machine which can also handle requests. Later if there is less workload this VM can be shut-down and the available resources can be used for other services. Figure 2.2 shows the dynamic resource allocation of cloud computing. As you can see the changing resource need is handled in real time in cloud computing environments and therefore the resource utilization is being increased [6].

Another advantage is, that the cloud computing user does not have to care about backing up data, load balancing and resilience planning and still has highly scaleable and highly available applications. Detailed information about the technical details are described in chapter .

Probably the only disadvantage of hosting applications in the cloud is, that you have to fully trust the cloud computing provider, because every system acts like a black box. This means that the user does not know where the applications and data are stored and how they are protected against unauthorized access. The data can be distributed in many data centers in many countries to enhance scalability and availability. For high sensitive data, (e.g. online banking systems) cloud computing with a public provider is obviously not a good choice.

Chapter 3

Technical Details

3.1 Virtualization

The most important concept of cloud computing and therefore also for IaaS systems is virtualization. Nowadays every cloud computing system is virtualized. This means that all the user's cloud computing applications run in virtual machines. In the case of IaaS systems the user buys a VM and can install whatever system he likes on it [1].

Virtualization brings many advantages. A lot of crucial concepts of cloud computing are a lot easier with virtualized systems. On example is load balancing: Virtual machines can be copied and can be migrated on other physical machines without any problems - Figure 3.1 illustrates this graphically. The following section will give you a greater insight into load balancing strategies [1].

3.2 Load Balancing

As mentioned in the previous section, the applications on cloud computing systems run in virtual machines, and these VMs run in physical machines. Furthermore, there is a huge variation on the load (or resource needs) on the applications, therefore if there run too many applications (or VMs) on one physical machine it may get overloaded. This is why one needs load balancing. Load balancing should avoid that the physical machines have to handle more resources than they

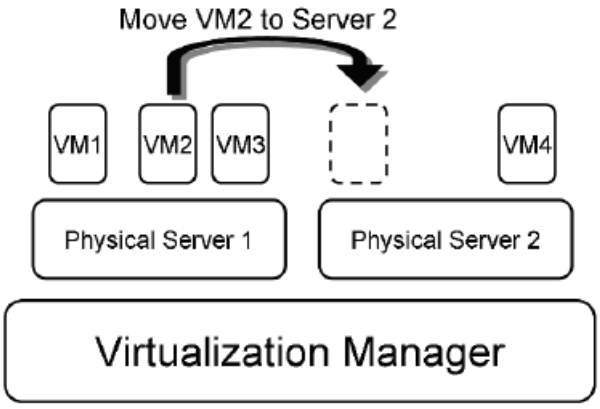

Figure 3.1: Virtualization and Live VM Migration in Cloud Computing Systems [1]

can offer. These resources can be CPU, RAM or storage space. If there run too many VMs on one physical machine this can mean a tremendous slow down of the VMs running on it. But slow VMs and applications are not the only problems, that bad load balancing would cause. If a application does not have sufficient resources, often the Service Level Agreement (SLA) is violated. A SLA is the agreement between customer and provider that guarantees certain parameters like performance and availability. A violation can mean that the cloud computing provider may have to give discount because of the violation or the customer decides to change to another cloud computing provider. As you can see good load balancing is crucial in the field of cloud computing [3].

3.2.1 Load Balancing Algorithms

There are many approaches on how load balancing algorithms should work, but they all have something in common: the input and the output. Every algorithm has to watch over the states of the physical and virtual machines and has to decide if a VM gets moved from. Furthermore, it has to decide which VM has to be migrated to which physical machine [3].

Many common load balancing algorithms are based on the current states of the PMs and VMs. They therefore are called ”reactive” algorithms. This means that when the resource utilization on a certain PM reaches a certain threshold, a VM gets migrated to another PM. This algorithm is rather easy to implement, as it

only has to watch the current resource utilization at the physical machines, but the disadvantages are that it only considers the current state of the system and that when a the threshold is reached most often, an imbalance situation is yet the case. Furthermore, it cannot guarantee a long-term balance situation, as it only acts on the parameters known at that point of time [1, 3].

Other algorithms are based on a "proactive" approach. In these algorithms the physical machines try to predict the resource demand of the VMs running on them and if in the near future there would be a overload they migrate a VM to another physical machine which predicts a lower resource utilization. This has the advantage that if the algorithm works as desired and the estimates on the resource demands of the VMs are approximately correct, there won't be any more overloads, because the algorithm would predict correct and migrate the VMs before the physical machine is overloaded. Usually the proactive approaches give better results than the earlier discussed reactive approach. But, however, there are also disadvantages. The first one is that the physical machine does under usual circumstances not know, which VM should be migrated to another PM. Furthermore, in the long run this approach also does not create a balanced state, because it only predicts a certain time span in the future. Finally, there have to be made a lot of calculations to predict the resource needs of the VMs (this is usually done by a Markov model), especially when there are a lot of VMs per physical machine. This often creates a big load just for the load balancing algorithm [2, 3].

3.3 Resilience Planning

3.4 Backup Strategies

3.5 Monitoring

Monitoring is an essential part for deployed systems for both, dedicated hardware structures and cloud computing systems. It enables to discover and analyze failure, bad configuration, performance bottle necks and many other issues that are important parts of software maintenance. Cloud computing systems are made

to be highly scaleable and elastic. But these two characteristics make it hard to monitor cloud computing systems as monitoring of high scaling systems requires to collect, store and analyze a lot of data in real time, which can be computationally highly expensive. Furthermore, the high elasticity shows an extra challenge, as the whole environment can change in a very short period of time in modern clouds [9].

3.5.1 Common Architectures for Distributed Systems

The most important architectures that are typical for monitoring distributed systems are:

- Flat Pull Model: A central server polls all the machines he should monitor according to a schedule when machines join and leave.
- Hierarchical Pull Model: Monitoring servers poll a subset of all the servers which should be monitored. A central server then polls the monitoring servers.
- Hierarchical Push Model: The server themselves push information to a monitoring server which is assigned to them and the monitoring servers collect that information from their machines and push it to a central server [9].

However, the just mentioned models for monitoring are modeled for grid and cluster computing. They are not always optimal for cloud computing systems as virtual machines can dynamically be added and removed and this creates the need for other approaches [9, 7].

3.5.2 Architectures modeled for Cloud Computing

In 2010, Huang [7] proposed a monitoring system he called "Push and Pull", which should combine the advantages of the push and pull. In current pull systems, the consumers (monitoring servers) pull information from the producers (virtual machines which should be monitored) in a certain interval. This approach is efficient

but lacks in consistency. If the interval is too big, data is lost, but if the interval is too small it is inefficient as there is too much network traffic. In push systems on the other hand, producers tell the consumers changes whenever the changes are greater than a threshold. This can produce good performance depending on the threshold and all data changes are monitored, but if the threshold is too small, too much information is transmitted and this leads again to inefficiency. The Push and Pull model Huang suggests, should use both approaches in a mixed way depending on the situation. Therefore, Huang introduces a value called User Tolerant Degree (UTD). This value indicates if high accuracy is a core requirement for the user or the if the user tolerates minor inaccuracy. In Push and Pull both approaches are used simultenously in different degrees. For example, if the UTD is small, the user won't tolerate inaccuracy and the push model will dominate over the pull model. This means that according to the UTD, the producer will push changes larger than a relatively small threshold to the consumer, but at the same time the consumer will pull with a long interval to be sure not to miss changes. But if the UTD is large (the user will tolerance inaccuracy), the pull model will dominate. The consumer will have a short interval in which he pulls information from the producer and the producer only pushes changes if they are bigger than a large threshold.

Another approach was proposed from Ward [9] in 2014. He called his system VARANUS and it is built up on a layered probabilistic multicast or gossip protocol. By dividing up computational complexity over the system, gossip protocols show great performance in large scale networks. Within the layers the peers communicate via push and pull with other peers which are near by. The distance between peers is determined by measuring the round trip time between the peers. In VARANUS the communication is different from layer to layer. In the lower layers, there is a great bandwidth and therefore the exchange rate is high and the information is highly consistent, whereas in the higher layers, the information is sent in lower intervals, because else there would be a too high network traffic. Figure 3.2 shows the architecture of the VARANUS.

Both architectures show significant improved results in terms of performance, latency and scalability in large scale deployment systems than the traditional systems for cluster and grid computing (discussed in section **??**).

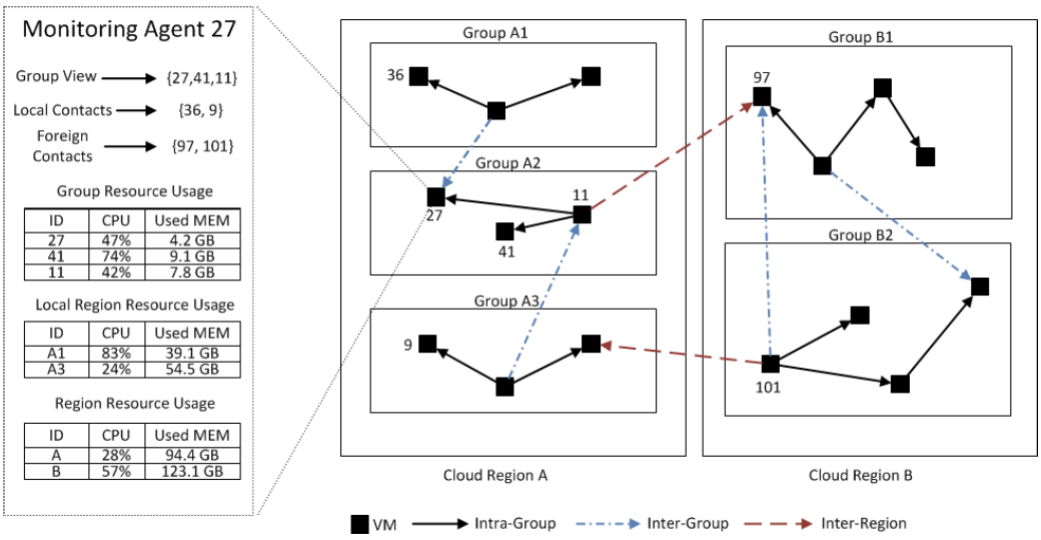

Figure 3.2: Architecture of the VARANUS monitoring system [9]

Chapter 4

Results and Discussion

We have already looked at the details of IaaS and how it works technically. Now it's time to give you some insights how such cloud infrastructures are implemented in real life and how resources can be provided and used by the consumers. Therefore we want to illustrate some concrete representatives who are are operating as a successful IaaS provider in practice in the following chapter.

4.1 Amazon Elastic Compute Cloud (Amazon EC2)

4.1.1 A first glance on Amazon's world

Amazon EC2 is part of the big Cloud Platform of the famous and globally well-known internet company Amazon.com, Inc., namely Amazon Web Services (AWS). AWS started offering IT infrastructure services to businesses using this Cloud computing model already in 2006. Today it's probably the most popular representative in this kind of IT sector offering highly reliable, scalable and low-cost infrastructure platform in the cloud used by a huge number of businesses in currently 190 countries around the world. They emphasize and try to implement the several benefits Cloud computing brings with it. Consumers should avoid hight initial costs to get their infrastructure running, instead a variable cost model with a pay-as-you-go price model should make their work much more cost-effective. Furthermore setting up and maintaining infrastructure components

should not be a common task businesses should care about. In contrast the focus should be set on the major issues regarding the concrete business and infrastructure capacity can be provided by AWS with just a click on-demand. AWS truly offers a broad variety of several different services distinguished by it's provided resource type. A typical key benefit someone often face when talking about and using AWS is the perfect interoperability of all the platform components comprising it. That's probably a major point what makes AWS so beloved by customers around the globe. Next to storage(Amazon S3) or networking (Amazon VPC) compute resources, managed by Amazon EC2 are essential to every business as these are the physical machines services need to be alive. [8]

4.1.2 EC2 Insights

Amazon EC2 is designed for web and system administrators to make their lives easier. It provides easy manageable compute capacity available in the cloud. Via a web interface the administrator can request more or less capacity with just a few clicks within minutes. Some more minutes later the required instances are fully set-up with the desires of the user and ready-to-go. This allows the user to scale up and down (which means renting more or compute capacity) on the fly depending on the current demand and without any need to configure them manually. To give this explanation further importance and motivate why Amazon EC2 is the right choice for so many businesses, we want to look at some key benefits this cloud technology introduces in contrast to buying just traditional dedicated hardware at the provider of your trust.

4.1.2.1 Elastic scaling

As already mentioned the customer can add and remove EC2 instances as she likes and therefore adapt the available compute capacity based on the current demand. It doesn't matter whether someone commissions just a few or even thousands of instances simultaneously. [3] Especially for companies whose resource demand is highly unpredictable due to the reason that they can't give a good guess about the future development of their product this EC2 technology can help them a lot. A question which probably arises is what happens if workload on the EC2 instances varies heavily due to the nature of the application. For instance there

could be the case that load heavily depends on some special days or time and always manually enlarging and shrinking the cloud infrastructure would be again an overhead. For this reasons Amazon introduced the Auto Scaling feature where the capacity dynamically scales up and down depending on the current resource demand. The user has opportunities to define peaks and troughs or specific time points where EC2 should react with scaling. So on the one hand you can ensure that there is always enough capacity to deal with the current load and on the other hand you can minimize your costs as you don't waste money for capacity you don't even need at the moment. Over the years Amazon developed some further possibilities for customers to get the most "bang for the buck". With the so called Reserved Instances they established a model where customers can agree to select one of the predefined instance types, pay some low initial costs for each instance they want to reserve and get a significant discount in variable costs. If you are not satisfied with your instances anymore you still have the possibility to place them in another AWS region, change the instance type or even sell capacity to other projects that end before the time frame for the Reserved instances expires. A further pretty nice instance model are Amazon EC2 Spot instances. With Spot instances the user can bid on the hourly price per instance she want to spend. Compute capacity again increases and decreases dynamically and as long as the bid meets or exceeds the instance price which customers commit to pay they gain access to them. This is a great possibility to be sure not to pay more than you actually want and prevent the risk that you oversee massive scaling. [8]

4.1.2.2 Elastic Load Balancing (ELB)

ELB is an intelligent solution to improve both scalability and fault tolerance. The load balancer component automatically distributes incoming traffic throughout the available instances to balance the load equally. It seamlessly integrates with the Auto Scaling feature described in the section above. You can still set your Auto Scaling conditions and if they are met instances get added to your Auto Scaling Group. This also works fine behind an Elastic Load Balancer. Achieving better fault tolerance is the second core point of using ELB. The Amazon load balancer automatically detects unhealthy instances behind itself and therefore does not distribute any further traffic to them until they get healthy again. So it's basically no problem when one of several instances is down for a while. Amazon lets you spread your EC2 instances within Availability Zones and multiple regions. A region is a specific geographic zone where Amazon operates and

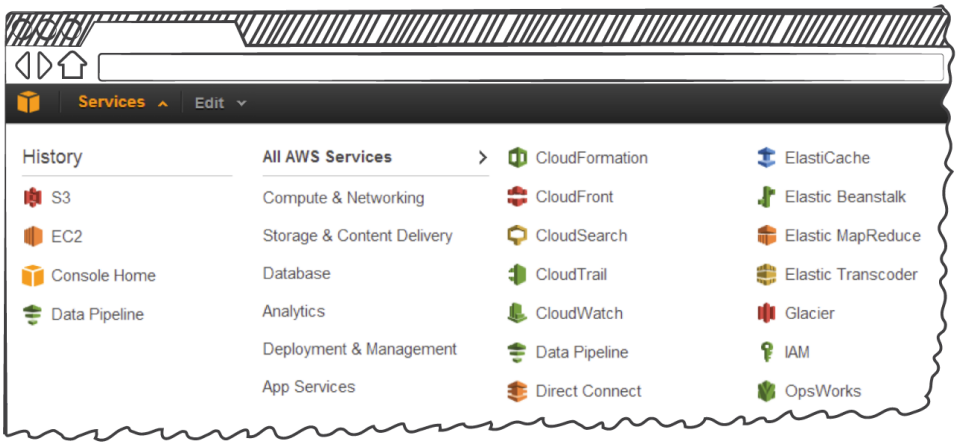

Figure 4.1: AWS Management Console - Main overview

contains multiple isolated Availability Zones. Especially for globally operating bigger businesses it is highly recommended to distribute instances throughout several Availability zones and multiple regions as this results in lower latencies and further increases fault tolerance. Although a complete availability zone is offline traffic can be routed to farther away regions intermediately. The nice integration with other Amazon Service such as the new Amazon Route 53 even enables DNS failovers in case the load balancer itself suffers.

4.1.2.3 AWS Management Console

Moreover what makes working with AWS so convenient is how less effort you need to complete such tasks. The AWS Management Console does not require much expertise in system or server administration as it comes with a comfortable graphical user interface along. There you can choose which of the dozens of services you would like to look at and also have root access to any of your EC2 instances and so you can start, stop, check and manage them as you like.

Chapter 5

Conclusions and Future Work

Bibliography

[1] Emmanuel Arzuaga and David R. Kaeli. Quantifying load imbalance on virtualized enterprise servers. In *Proceedings of the First Joint WOSP/SIPEW International Conference on Performance Engineering*, WOSP/SIPEW '10, pages 235–242, New York, NY, USA, 2010. ACM.

[2] A. Beloglazov and R. Buyya. Managing overloaded hosts for dynamic consolidation of virtual machines in cloud data centers under quality of service constraints. *Parallel and Distributed Systems, IEEE Transactions on*, 24(7):1366–1379, July 2013.

[3] Liuhua Chen, Haiying Shen, and Karan Sapra. Distributed autonomous virtual resource management in datacenters using finite-markov decision process. In *Proceedings of the ACM Symposium on Cloud Computing*, SOCC '14, pages 24:1–24:13, New York, NY, USA, 2014. ACM.

[4] Dialogic Corporation. Introduction to cloud computing, jul 2010.

[5] Oracle Corporation. Making infrastructure-as-a-service in the enterprise a reality, apr 2012.

[6] A. Gajbhiye and K.M.P. Shrivastva. Cloud computing: Need, enabling technology, architecture, advantages and challenges. In *Confluence The Next Generation Information Technology Summit (Confluence), 2014 5th International Conference* -, pages 1–7, Sept 2014.

[7] He Huang and Liqiang Wang. P amp;p: A combined push-pull model for resource monitoring in cloud computing environment. In *Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on*, pages 260–267, July 2010.

[8] Sajee Mathew. Overview of amazon web services, nov 2014.

[9] Jonathan Stuart Ward and Adam Barker. Self managing monitoring for highly elastic large scale cloud deployments. In *Proceedings of the Sixth International Workshop on Data Intensive Distributed Computing*, DIDC '14, pages 3–10, New York, NY, USA, 2014. ACM.