

An Applied Evaluation and Assessment of Cloud Computing Platforms

Daniel Högberg

January 21, 2012

Master's Thesis in Computing Science, 30 credits

Supervisor at CS-UmU: Mikael Rännar

Examiner: Fredrik Georgsson

UMEÅ UNIVERSITY
DEPARTMENT OF COMPUTING SCIENCE
SE-901 87 UMEÅ
SWEDEN

Abstract

Cloud computing is an emerging paradigm with the potential to change the way computing resources are used by enabling the long held idea of utility computing. This thesis aims to conduct a survey of the cloud computing platforms that are currently available and to compare and evaluate the alternatives. Criteria that are important to consider when choosing between cloud platforms are defined and used to compare a set of selected platforms. A case management application called Wera is also migrated to platforms to test the migration processes and the platforms in practice.

An experience gained from performing migrations to several Infrastructure-as-a-Service platforms is that they are very much alike. The storage models and features available may differ but the functionality offered is essentially the same. The fact that the area is still new is very visible when working with the platforms, but even though the platforms are still evolving, they are useful. Disruptions in the availability are rare and it is surprisingly easy to migrate an application to an Infrastructure-as-a-Service platform and have it run in the cloud. Employing Platform-as-a-Service offerings requires a greater effort to get started but using them there is even more to gain by tasks like patching and automatic scaling being transferred to the provider.

Contents

1	Introduction	1
1.1	A brief overview of Sogeti	1
1.2	Thesis outline	1
2	Problem Description	3
2.1	Goals and Purposes	3
2.2	Method	3
2.2.1	Finding Candidate Cloud Platforms	4
2.2.2	Defining Evaluation Criteria	4
2.2.3	Performing migrations	5
3	Cloud Computing - an Overview	7
3.1	Defining Cloud Computing	7
3.1.1	Definition I	8
3.1.2	Definition II	8
3.1.3	Definition III	9
3.1.4	Conclusion	9
3.2	Motivators for Using Cloud Computing	10
3.3	Objections to Cloud Computing	11
4	Accomplishment	15
4.1	Preliminaries	15
4.2	Evaluation	16
4.3	Migrations	16
5	Cloud Computing Platforms	19
5.1	Amazon Web Services	19
5.1.1	Elastic Compute Cloud	20
5.1.2	CloudWatch	20
5.1.3	Virtual Private Cloud	20
5.1.4	Storage	21
5.2	Atlantic.Net	22

5.3	City Cloud	22
5.4	Flexiant Flexiscale	23
5.5	GoGrid	23
5.6	Google App Engine	24
5.6.1	The Platform	24
5.6.2	The Datastore	24
5.6.3	Billing	25
5.6.4	App Engine for Business	25
5.7	Joyent	25
5.8	Microsoft Azure	26
5.8.1	Windows Azure	26
5.8.2	SQL Azure	27
5.8.3	Windows Azure AppFabric	27
5.8.4	Windows Azure Marketplace	29
5.9	OpSource	29
5.10	Rackspace	29
5.10.1	Cloud Files	29
5.10.2	Cloud Servers	30
5.11	Salesforce Force.com	30
6	App Engine vs. Azure	31
6.1	Level of Virtualization	31
6.2	Storage	32
6.3	Support for Application Types	32
6.4	Software Development	32
6.5	Related Studies	33
7	Migrations	35
7.1	Infrastructure-as-a-Service Platforms	35
7.1.1	Registration Process	36
7.1.2	Snapshots	36
7.1.3	Load Balancing	38
7.2	Microsoft Azure	38
7.2.1	Migrating to SQL Azure	39
7.2.2	Migrating to Windows Azure	40
7.3	Migrating to Google App Engine	41
7.3.1	The Users Service	42
7.3.2	The Datastore	42

8	Evaluation of Platforms	45
8.1	Cost	45
8.2	SLA	47
8.3	Security	48
8.4	Support and Documentation	50
8.5	Complexity of Migration	51
8.6	Results	51
9	Conclusions	53
9.1	Limitations	54
9.2	Future Work	54
10	Acknowledgements	55
	References	57
A	Cost Evaluation	61
B	SLA Evaluation	63
C	Support and Documentation Evaluation	65
D	List of Abbreviations and Acronyms	67

List of Figures

3.1	The cloud computing stack.	8
7.1	Set-up with multiple web servers.	38
8.1	The cost of running the two scenarios on the platforms in SEK/month. (Exchange rates from 10/06/2011 when 1£ bought 10.26SEK and 1\$ bought 6.32SEK)	46
8.2	The weights assigned to the categories for Wera.	52

List of Tables

7.1	The differences between EBS- and S3-backed AMIs	37
8.1	The score awarded for the SLAs.	48
8.2	The points awarded to the platforms for support and documentation.	50
8.3	The points awarded to the platforms in each of the categories.	52
8.4	The points awarded to the platforms calculated with the assigned weights for Wera.	52
A.1	The cost for the services, the currency is USD unless otherwise stated. Instance 1 corresponds to at least 2 GB ram and 2 CPUs and Instance 2 at least 4 GB ram and 4 CPUs.	61
A.2	The cost for the services, the currency is USD unless otherwise stated.	62
B.1	The SLA guarantees offered by the vendors, hardware and network percentages are the promised uptime for each month (Amazon measures uptime by the year), and the compensation percentages are based on the monthly cost. Min. dur. is the minimum duration for a failure that grants any compensation.	63
B.2	The compensations offered by vendors for hardware and network failures of 10 minutes, 1 hour, and 10 hours. The percentages are the amount of the monthly bill refunded.	64
C.1	The support offered for the platforms.	65
C.2	A rating of FAQ, forum, wiki and other information resources for the platforms. Note that these ratings are subjective and based on my personal opinions.	66

Chapter 1

Introduction

Cloud computing is one of the big trends in the IT industry today. The area is still new and constantly growing and many companies are adapting to the paradigm, both as suppliers and consumers. The service offered by suppliers varies but they all provide remotely accessible computing resources. These resources can be servers, storage, applications and more. The advantage of cloud computing is that an organization that wants to host an application of some sort can do this without having to buy and maintain hardware and software of their own. Hosting the application in the cloud also means that the capacity can be adapted to the current load and in this way an increase or decline in the number of users can be handled.

This master's thesis aims to compare and evaluate different cloud platforms. An existing application will then be migrated to the most interesting and suitable platforms to test the platforms and the migration process in practice. This will result in a recommendation whether or not to migrate the application to the cloud and the choice of platform. An additional objective of the thesis is to contribute to the knowledge of cloud platforms at Sogeti who are interested in learning more about the cloud computing alternatives available on the market.

1.1 A brief overview of Sogeti

Sogeti is a consulting company offering IT services, for example IT specialist services, developing projects, local system management, and management and development of the customer's IT infrastructures. Sogeti is present in many locations in Sweden and the world. In Umeå the company specializes in the following businesses:

- Banking and finance
- Public sector
- Forest and paper industry
- Manufacturing and processing industry

1.2 Thesis outline

- Chapter 2 gives an introduction to the problem and lists goals and purposes.

- Chapter 3 provides an introduction to the term cloud computing and what it entails. Motivators for using the paradigm and challenges that exist are presented.
- Chapter 4 describes the working process of the project, some problems encountered, and changes made.
- Chapter 5 gives a description of the cloud computing platforms that have been evaluated.
- Chapter 6 compares the two Platform-as-a-Service offerings Google App Engine and Windows Azure in detail.
- Chapter 7 describes the processes of migrating Wera to cloud computing platforms.
- Chapter 8 presents the results of the comparison of cloud computing platforms.
- Chapter 9 contains a discussion on the thesis and presents conclusions drawn.
- Chapter 10 lists acknowledgements.

Chapter 2

Problem Description

The main tasks of this project are:

- Conduct a survey of the market of cloud computing platforms and select possible candidates for migration
- Define criteria for evaluating the chosen platforms and compare them according to these criteria
- Construct test cases to verify the functionality of the application on the platforms
- Migrate an application to the chosen platforms and compare the migration processes

The application to be migrated is a case management application called Wera. It has been developed by Sogeti and is currently used by the company in customer projects. The application is implemented with Visual Basic .NET and uses an SQL Server database.

2.1 Goals and Purposes

The objective of the thesis is to define criteria that are important to consider when choosing between cloud platforms and to apply these criteria on available platforms to find a suitable platform for Wera. The project should also result in a conclusion as to whether it is beneficial to migrate Wera to the cloud or not. It should also contribute to the knowledge of cloud computing platforms at Sogeti.

Goals:

- An evaluation of 6-15 cloud computing platforms based on relevant criteria
- Migration of Wera to at least two cloud computing platforms
- Recommendation for migration of Wera based on the conclusions of the evaluation and migrations

2.2 Method

The following methods will be used:

- Information gathering
- Evaluation (according to specified criteria)
- Practical testing

The first task of the project will be to find candidate platforms for the evaluation. Then the criteria for the evaluation and the way to compare the different solutions will be defined. The evaluation will thereafter be performed and result in a ranking of the platforms and the most interesting platforms will be subject to more research and migration.

2.2.1 Finding Candidate Cloud Platforms

There are some basic demands that platforms have to meet to be taken in consideration for this project. Since Wera is developed in .NET Infrastructure-as-a-Service vendors has to offer the ability to run Windows images. Platforms also have to comply with the cloud computing properties defined in 4.1.

To find possible candidates for the evaluation the market will be surveyed for established cloud vendors. A review of the market by Gartner, the "Magic Quadrant for Web Hosting and Cloud Infrastructure Services (On-Demand)" [32], will also be used to find candidates.

2.2.2 Defining Evaluation Criteria

There are many things to consider when deciding on what cloud platform to use when migrating an application. According to [34] important things to consider are "level of customer support, security and service level provided as well as cost and ability to scale". Of these properties, all are relevant for Wera except possibly for the ability to scale. Since one of the objects of this project is to contribute to the knowledge of cloud platforms at Sogeti, the concerns of *their* customers are important to consider as well. Complexity of migration is one such property.

In view of this the criteria that the evaluation will be based on are:

- Cost - calculated on normal usage of Wera
- SLA (Service Level Agreement), promised availability
- Security, confidentiality and authentication
- Support and documentation
- Complexity of migration

Most of the criteria above are hard to quantify and measure according to some scale. Therefore the platforms will be compared to one another and given a rating for each of the criteria. These ratings will correspond to a score and the different criteria will be weighted based on importance resulting in a total score for each platform.

The cost of running an application on a platform can be difficult to predict. Some vendors have complex pricing models, the number of clock cycles on a CPU that will be consumed or the amount of inbound and outbound traffic that will be produced are impossible to predict exactly. In spite of this a normal level of usage for Wera will be defined and the cost for the platforms approximated.

A SLA states what is promised by a service, for example what degree of uptime can be expected, and how a customer is compensated if the SLA is not met. Therefore, the SLA helps in evaluating the quality of a service. However, if the system becomes unavailable, the consequences for the customer are likely more severe than the compensation offered makes up for. However, the most commonly occurring parts of the SLAs will be compared and rated according to the score system. More specific promises will also be mentioned and if deemed relevant grant extra points.

Cloud computing introduces new security issues and the area of the security concerning cloud computing is a large subject. In this study the information made available by each vendor will be compared and openness about the techniques used is an important aspect so that the security of platforms can be assessed.

When it comes to the support offered by vendors; availability and fast responses are of great importance since problems causing downtime of services can be very costly. Good documentation also helps reduce cost when migrating to a platform. The rating will be based on the level of support offered and the quality and extent of the documentation.

The ease of migrating an application to a platform is another important aspect since a smooth process will save a lot of time. The complexity of migration can only be estimated in the theoretical evaluation but some indications can offer information on the complexity of different solutions. The migration scenarios that will later be carried out will be evaluated in more detail when the actual information is acquired.

The area is still new and many platforms may not have been tested thoroughly. Third party reviews are hard to find. Therefore the main source of information is the vendors themselves in many cases. A critical attitude towards the material will be applied.

2.2.3 Performing migrations

The plan for the process of migrating Wera to new cloud computing platforms is composed of the following steps:

- Information gathering on how to deploy a system on the platform.
- Practical work.
 - Provision an instance on the platform.
 - Migrate the application.
 - Run test cases.
- Additional research to configure the system or to solve problems that may occur.

Chapter 3

Cloud Computing - an Overview

To have a meaningful discussion about any subject it is important that all participants agree upon a definition of the subject of discussion. This has not been the case recently when the subject at hand is cloud computing or "The Cloud". The cloud has become a buzzword that has been used as a label for many products where it may not belong which has made the term fuzzy. The lack of clearness surrounding the term may also stem from the fact that cloud computing encompasses many techniques and different service models. This chapter will attempt to clarify what cloud computing is, what benefits it may offer, and what the disadvantages are.

3.1 Defining Cloud Computing

Although attempts have been made, there is no universally accepted definition of what is and is not cloud computing. However, there is a commonly used classification of service models that differentiates offerings based on the level of virtualization; the three most frequently mentioned models are Software-, Platform- and Infrastructure-as-a-Service [36], [41], [35], [43].

Software-as-a-Service (SaaS) is a piece of software made available to customers through the Internet. The application runs on the infrastructure of a cloud provider and is typically accessed by users through a web browser. The users of the service are not concerned with the installation of the software or keeping it up to date and are often billed by the user months or according to some other pay-as-you-go pricing plan.

Platform-as-a-Service (PaaS) allows for the user to develop applications with the programming languages and tools made available by the platform supplier. These applications can then be run on the infrastructure of the platform which the user does not manage or control directly.

Infrastructure-as-a-Service (IaaS) customers are provided with the capabilities to provision computing resources like processing, storage, and networks. These resources can be used to deploy and develop arbitrary software and the customer is fully responsible for the administration of the operating systems and other software installed. Figure 3.1 visualizes this cloud computing stack.

Here follows an overview of some of the attempts to formalize cloud computing.

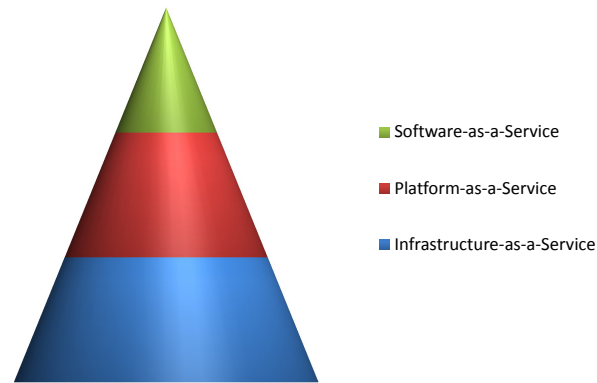


Figure 3.1: The cloud computing stack.

3.1.1 Definition I

In the article *A Break in the Clouds: Towards a Cloud Definition*, Vaquero et al. [41] aims to achieve a complete definition of what a cloud is, by using the main characteristics associated with the paradigm. Numerous definitions have been reviewed to form a consensus definition and to extract the essential characteristics forming a minimum definition. The three service models mentioned above are presented as the scenarios where clouds are used.

Clouds are a large pool of easily usable and accessible virtualized resources (such as hardware, development platforms and/or services). These resources can be dynamically reconfigured to adjust to a variable load (scale), allowing also for an optimum resource utilization. This pool of resources is typically exploited by a pay-per-use model in which guarantees are offered by the Infrastructure Provider by means of customized SLAs.

No single feature were proposed by all the reviewed definitions but the minimum definition of essential characteristics were *scalability, pay-per-use utility model, and virtualization*.

3.1.2 Definition II

Another definition is provided by NIST, National Institute of Standards and Technology, in [35], it reads:

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.

The NIST model of cloud computing is also composed of five essential characteristics, three service models, and four deployment models. The essential characteristics are:

- On-demand self-service - It is possible for a user to provision resources at any time without human interaction.
- Broad Network access - The resources are available through a network to various standard platforms.

- Resource pooling - The resource pools of the provider are serving multiple customers and can be dynamically assigned as demand changes.
- Rapid elasticity - The customer can rapidly provision more resources or release some of the provisioned resources at any time.
- Measured Service - The usage is automatically metered at some level of abstraction to provide a transparent usage reporting for both the user and the provider.

The three service models included are Software-, Platform-, and Infrastructure-as-a-Service as mentioned above and the deployment models suggested are:

- Private cloud - the infrastructure is not shared outside of the organization employing the private cloud and is managed internally or by a third party.
- Community cloud - the infrastructure is shared among organizations with some common concerns like security requirements, policy, and compliance. May be managed by the organizations or a third party.
- Public cloud - the infrastructure is made available to the public and is owned by an organization selling cloud services.
- Hybrid cloud - the infrastructure is a composition of two or more clouds of the above variations. The clouds are unique but there is data and application portability between them.

3.1.3 Definition III

In the article *Above the Clouds: A Berkeley View of Cloud Computing* Armburst et al. [18] defines a *Cloud* as the datacenter hardware and software used for hosting and if it is made available to the public with a pay-as-you-go pricing model, it is a *Public Cloud* and the service sold *Utility Computing*. *Cloud computing* is defined as SaaS and Utility Computing but excludes Private Clouds.

The authors refrain from using X-as-a-Service terms since they could not reach an agreement as to what exactly differentiated the terms. However, SaaS is used since it has been used for a long time and has well understood advantages. Instead of the XaaS model, different cloud offerings are distinguished based on the level of abstraction presented to the programmer and the level of management of the resources provided. Examples are given of the computation-, storage-, and networking model for Amazon EC2, Microsoft Azure, and Google App Engine. The computation model for EC2 can run practically any application or operating system since it is based on the x86 architecture via Xen virtual machines. Although Azure is suited for general purpose applications as well, the virtualization level is raised to the Common Language Runtime virtual machine (today, lower levels of the Azure platform can be accessed as well). App Engine raises the virtualization level further and is specialized for request-reply based web applications. The applications have to adhere to a predefined structure and use the frameworks provided for Java and Python.

3.1.4 Conclusion

Definition I and II although not equivalent does not contradict each other. Definition III is less expressive regarding what cloud computing means but chooses another model for classification of cloud offerings by excluding the use of XaaS terms. This approach is

supported in [27] where the author analyses cloud offerings along various dimensions of comparison and argues that such analyses gives a more revealing result than using one-dimensional categorization of PaaS and IaaS.

This thesis is focused on public IaaS and PaaS offerings, SaaS is therefore not considered in the concept of cloud computing throughout this report.

3.2 Motivators for Using Cloud Computing

Advantages of using cloud computing include:

- Pay-as-you-go pricing
- Elastic scalability
- Possibility to scale "infinitely" large
- Focus on business advantage
- "Cost associativity"

One major motivator for using cloud computing is the economic aspect and the potential savings. The pay-as-you-go model of cloud computing entails exchanging capital expenditures for operational expenditures and the economies of scale enjoyed by large corporations delivering cloud services is a potential cost saver for cloud users. The need to make investments in hardware to create a private datacenter for hosting is eliminated by the cloud model which is particularly advantageous for startups lacking the funds to make such investments.

The elasticity of cloud computing can also be a great source of monetary gain. With a private datacenter the server capacity for expected load need to be provisioned in advance since delivery, installation and configuration of new hardware takes time. Predicting future load is difficult, [18] gives the example of a company that made its service available on Facebook and had an increase in demand from 50 servers to 3500 servers in three days. Even though not all companies are likely to experience such surges in demand under- or over-provisioning are still risks. Over-provisioning results in unnecessary expenses for hardware that is not needed. The consequences of under-provisioning are likely even worse since insufficient capacity will result in a poor user experience that may result in a loss of customers. Failing in the provisioning of the hardware leads to these undesired scenarios but even if the prediction is correct and the hardware can handle the peak load the capacity is likely unnecessarily high the majority of the time. The load on many systems varies over the course of each day, week, or year. The load during nights, weekends, or a particular season may be lower than the average so even successfully provisioning for peak load may lead to a lot of unutilized capacity. In fact, an estimate of utilization of server capacity in datacenters puts the number at 6 % [37]. This illustrates the benefit of being able to quickly scale up and down according to current load.

The headline of the Microsoft Azure site is currently "Focus on your application. Not the infrastructure."¹ and this is a frequently used argument for the use of cloud computing. Indeed many corporations need a datacenter of their own for managing their systems even though IT is not their primary expertise. Not all such corporations may be able to handle the constant progress and expansion in the field leading to systems growing out of control. On a cloud platform the systems and infrastructure are taken care of by specialists and a

¹<http://www.microsoft.com/windowsazure/> 2011-06-08

company whose core area of competence is just the management and administration of IT infrastructure.

An advantage with platform level offerings is that the complete software stack is managed by the provider. The user is relieved of the tasks of installation, configuration, and patching of the operating system and other software. Additionally, the need for software licenses is reduced or eliminated since this is included in the platform.

Billing per computation hour makes the service cost associative, the price for using 1000 instances for one hour is the same as using one instance for a thousand hours. This can be a great property for a user with a large computationally heavy and easily parallelized task where the result is wanted as fast as possible. An example of this is [42], when 17,481 pages of Hillary Clinton's travel documents were released an engineer at Washington Post converted them to a machine searchable format in nine hours using just over 1400 instance hours on Amazon EC2. However, a word of caution is in place here since Amazon, for example, charges for each started instance hour even if the instance is terminated after a few minutes. This means that although using 1000 instances for one hour does cost the same as using one instance for a thousand hours, using 1000 instances for 61 minutes cost twice as much.

3.3 Objections to Cloud Computing

There are many objections to cloud computing and opinions on why it should not be used. Mostly, the same arguments are used repeatedly. The objections may be real problems but some can be overcome or may not be specific to cloud computing. Some of the most common objections to cloud computing are [11], [18], [31]:

- Availability
- Lock-in
- Software licensing
- Legal issues
- Performance
- Security

The availability is very critical in many systems and the sense of losing control may discourage migrations to cloud computing platforms. It is a fact that even the large providers have had occurrences of downtime in different systems, the most recent big outage happened in Amazon's datacenter in North Virginia[26]. A failure in the data center caused a "remirroring storm" where disks in the Elastic Block Storage started to create replicas using up all of the capacity. The failure was not completely contained within the affected Availability Zone, which according to Amazon are without any common single points of failure. For some users, the outage resulted in days of downtime. However, outages are still rare occurrences and the question is how robust and secure current on-premises infrastructure and systems are. An analogy describing the situation compares the cloud to an airline, if it crashes that will affect many but it is still a safer means of travel than a car, statistically. In [40] the recommendation is to try and negotiate organization-specific terms where the measures of success are part of the payment structure and clear penalties are agreed upon in case of failure to deliver. In case this is not possible the advice is to carefully choose which workloads

to use cloud computing for. For systems demanding really high availability, the authors of [18] believes that the only solution is to use multiple cloud providers. However, this calls for interoperability between platforms and incurs extra complexity to the migration.

Lock-in is a common concern when considering a migration to a cloud platform, it is mentioned in all of [18], [40], and [27]. The concept applies to both data and applications and the concern is that the investments made in a platform may be lost if for some reason the platform can no longer be used. There can be several reasons for a user to want to leave a platform. The price, service quality, or some other characteristic of the platform may change, or in the worst case, the supplier might go out of business. In such an event, the user will need to move the data and applications to be hosted somewhere else and how easily this can be done is a large concern. There are not yet any standard APIs used among platforms to facilitate interoperability, although attempts for this are being made². Migrating an application from one platform to another will require some amount of work, how much is dependent on the actual platforms. PaaS solutions are by nature more tied to the corresponding platform but IaaS solutions may have special features that only exist on one platform making migrations cumbersome in both cases. Data lock-in may be a problem as well; it is unclear how migrations of large datasets from cloud based data stores can be performed. Even if application and data can be migrated out of a platform, an alternative with the same capacity to scale or performance may not exist.

Software licensing is listed as an obstacle to the growth of cloud computing in [18]. The software license model of buying a certain number of licenses for running the software on a fixed number of machines is not very suitable for cloud computing where the number of instances may grow and decline over time. Current software licenses may not even apply for use on cloud platforms. An example of this is Amazon EC2 where instances running Windows Server cost more per hour than Linux instances, in the FAQ of EC2 it is stated that Microsoft Windows Server licensing does not currently support using an existing Windows license in Amazon EC2 or any other cloud environment [5].

One obstacle for the adoption of cloud computing is compliance and legal issues. Different countries have different legislations and laws may forbid that certain types of data are stored outside of a country or other region like the European Union. This may be out of the users control on cloud platforms prohibiting the use of these platforms for some users. Personal records in particular have laws that regulate the handling of the data and these laws may not be suited for the cloud computing model. Since the area is still relatively new it may not be clear what rules apply in some situations. An example of the importance of the geographical location of services is given in [27]: "the location of a cloud provider's data may be critical for legal reasons; for example, Canadian universities cannot use Google's email hosting service due to regulatory concerns over conflicts with Canadian academic privacy laws and the USA PATRIOT Act."

Since cloud computing providers use virtualization to run many instances on the same physical server, performance unpredictability can be a problem. The results of a benchmark performed on 75 EC2 instances are presented in [18]. The mean memory bandwidth was 1355 MB/s and the standard deviation was 52 MB/s, less than 4 % of the mean suggesting that the instances could share the memory bandwidth quite well. For disk writes the mean bandwidth was 55 MB/s and the standard deviation 9 MB/s, more than 16 % of the mean suggesting that the disk I/O performance is more of a problem for the virtual machines. Another potential problem is that a provider can over-provision the total capacity affecting the performance negatively. The platform providers rarely give any guarantee regarding performance. In section 8.2, the service level agreement of many providers are studied in

²<http://www.opencloudmanifesto.org/> visited 2011-06-19

detail and none of them specifies any performance guarantees for the virtual machines.

High performance computing is a possible scope of use for cloud computing, there are areas where the cost associativity described above can be of value like movie animation or financial analysis. Such applications are implemented to run in parallel systems using for example the message-passing interface MPI, making cloud computing platforms well suited for running the applications. A problem that exists on cloud platforms is that the programmer cannot ensure that all threads of the application are running simultaneously. This may be a problem because some distributed applications use barriers of synchronization where the threads communicate after a phase of computation. If some threads are not allowed to run instantly, the rest of the threads will have to wait impacting the total performance. In [18], the use of some "gang scheduling" mechanism is suggested to solve this problem.

If an application or a system is moved to a cloud computing platform, the latency and transfer rate in and out of the platform may become a problem. Many applications today manage large datasets and low transfer rates may cause transfers to take a very long time. A possible solution for a user who wish to perform computations on large datasets using a cloud platform but also wants to avoid long transfer times is given in [18]. An example is presented to illustrate: A user wants to send 10 TB of data from San Francisco to Amazon located in Seattle. The bandwidth is measured to 20 Mb/s, giving a transfer time of 45 days. The charge for the transfer would be \$1000. The alternative is sending the data on ten 1 TB disks with a delivery firm. The transfer time would in this case be less than a day including shipping and unloading time and the time required for Amazon to load the data into the system. The cost for this alternative is estimated to \$400 for the shipping to Amazon, labor cost of Amazon, and shipping back. The cost of the disks is not included. This approach certainly does not feel like the data transfer method of the future but clearly has the potential to reduce the impact of transfer rate problems for large data sets. The authors also note that in the last five years, the cost per GB of disks had decreased by a factor of ten while the cost of wide area links decreased by a factor of three, indicating that the alternative of shipping disks will get more attractive over time.

Security is the most common objection to cloud computing but not a very specific one. The view in [40] is that security analyses of platforms are warranted but that there are many widely tested technical solutions that can be employed. The security of communication channels, encryption of stored data are examples of areas where well known techniques can be used. Another point made is that studies regarding security breaches show that a considerable number of the breaches are carried out from the inside of the organization, suggesting that the cloud could make the IT more secure in this regard.

In [27] it is stated that the security evaluation of the platforms available will mostly depend on the reputation of the providers and their actual results in practice, admitting that security breaches may not be disclosed to the public. The advocacy for this approach to the evaluation of security stems from the opinion that security is notoriously hard to quantify or compare qualitatively.

In [43], data security is listed as a research challenge in cloud computing. The providers must be able to provide *confidentiality* and *auditability* to ensure safe data access and storage and that security settings have not been tampered with. Confidentiality can be achieved using cryptographic protocols and auditability with remote attestation techniques. The trusted platform modules that can be used for remote attestation will need to be adapted to handle the dynamic migration of virtual machines between physical machines.

The topic of security is listed as an obstacle for cloud computing in [18] as well. However, the belief of the authors is that there is no fundamental obstacle to making a cloud platform as secure as the majority of conventional on-premise IT environments. Many of the obstacles

can be overcome with technologies such as encrypted storage, Virtual Local Area Networks, and firewalls, etc.

Chapter 4

Accomplishment

This chapter will describe how the work proceeded, from the selection of platforms for the study to the evaluation and the migrations to platforms.

4.1 Preliminaries

To find possible candidates for the evaluation the following approach was used. First the cloud computing market was surveyed. Market leaders such as Amazon, Google, Microsoft, Rackspace and Salesforce were included because they can be seen as indicators when judging the quality of other platforms. Amazon EC2 in particular but also Rackspace Cloud Servers has become platforms that many newly formed and less known platform providers compare their products with¹. Then, the "Magic Quadrant for Web Hosting and Cloud Infrastructure Services (On-Demand)" by Gartner was used to find more candidates. Gartner is an information technology research and advisory firm and the magic quadrant gives a graphical representation on Gartner's view of the "ability to execute" and the "completeness of vision" of different cloud vendors [32]. Since Sogeti estimated that customers could be interested in Swedish cloud service providers such alternatives were researched as well. To finish off the search for candidates, the Internet was also searched to find additional cloud service providers.

As mentioned earlier the word "Cloud" gets put on many products to make them more appealing even though the typical cloud characteristics may be missing in the offerings. Since the focus of this thesis is on cloud platforms the following characteristics were sought after in the platforms considered.

- The signup process should be possible to carry out online in an automatic fashion.
- The billing granularity should be fine, typically, servers should be billed by the hour and not by the month.
- There should not be any long term commitment associated with the service.
- The service should be able to rapidly scale up and down, automatically or through administrative interfaces.

¹OpSource, GoGrid, and Joyent are examples of this.

4.2 Evaluation

Initially the platforms were researched one by one to gain a general knowledge of the different offerings. Then a more focused examination of the platforms regarding each of the criteria was performed.

In order to predict the cost of running an application, likely usage scenarios had to be created. Two different migration scenarios have been used as base for the calculations. The first scenario is one instance running the database and another running the web server and the second scenario is one single instance running both. Minimum technical specifications for the servers were defined based on the requirements set by the suppliers of the software used. An approximation of the usage of Wera was also outlined in collaboration with my supervisor.

The SLAs has been evaluated in three categories: compensation for hardware outages, compensation for network outages, and whether the SLA includes scheduled downtime. The compensation for downtime is considered more important and therefore has been weighed the highest. The compensations have been compared with respect to downtimes of ten minutes, one hour and ten hours.

To rate the security of the platforms the plan was to review the information on the subject supplied by the providers since no third party source of information was found. This turned out to be very difficult since the focus and extent of security information provided is very different among the platforms. A more advanced method of actually testing different aspects of the security would have been needed to produce a tangible result. A compilation of the information offered by the suppliers can be read in section 8.3.

To rate the support and documentation of platforms the level of support offered has been investigated and the available documentation has been reviewed. The support rating for the providers has been based on the availability of the support and diversity in support channels. The documentation of the platforms has been reviewed to find differences and similarities. FAQs, forums and wikis have been rated based on comprehensiveness, activeness and clearness. Points have also been awarded for other resources that may be of value to customers.

The original plan was to estimate the complexity of a migration to each platform and to migrate the application to two or three platforms. It turned out to be difficult to make such estimations based on the information available and therefore this criterion was held off until the practical part of the project. It then turned out that some migrations were less time consuming than expected. Because of this migrations were carried out to nearly all of the IaaS platforms in the evaluation to gain practical experience from as many platforms as possible.

4.3 Migrations

After the evaluation had been carried out the first platform chosen for migration was Amazon EC2. After some research on setting up instances and creating snapshots using the elastic block store Wera was successfully migrated to the platform. The process of getting the application up and running on the platform went faster than expected.

The next platform to migrate to was another infrastructure service, City Cloud. Much of the experience gained from the Amazon migration was useful when working with City Cloud since some steps of the process were the same. Wera was again migrated without any major problems.

After these two the attention was turned to Microsoft Azure. Being a PaaS offering the migration to Azure was very different. The first step was to migrate the database to SQL Azure which was done with the help of a tool called SQL Azure Migration Wizard. The database could be migrated with only some slight modifications which were necessary due to some restrictions of the service. There is support for converting .NET applications to web roles that run on Azure but the standard process was not enough to get the application to work on the platform. The application had to be somewhat modified to use Azure Storage instead of the local file system and web service parts of Wera had to be set up as virtual applications. Solving these issues took some time and effort but eventually the application was working correctly on the platform.

Once the Azure migration was completed the initial goal of performing two to three migrations was fulfilled. Since the migrations to infrastructure services were fairly quick it was decided to try out most of the ones in the evaluation to get a broader experience of different offerings. Therefore Wera was migrated to Rackspace, GoGrid, OpSource, and Flexiant.

Finally Google App Engine was tested as well to gain experience from a platform that differs a lot from the rest. Since the platform only supports Java and Python, the original application could not be used. Instead a more simple proof of concept version of a case management application was developed in Java and deployed to App Engine.

Chapter 5

Cloud Computing Platforms

The cloud platforms to be evaluated are:

- Amazon EC2
- Atlantic.Net
- City Cloud
- Flexiant Flexiscale
- GoGrid
- Google App Engine
- Joyent
- Microsoft Azure
- OpSource
- Rackspace
- Salesforce Force.com

AT&T and Verizon were first supposed to be in the evaluation but have been excluded because their cloud services are only available in the U.S. Below follows a presentation of the vendors and a description of the corresponding platforms.

Amazon EC2, City Cloud, Flexiant Flexiscale, GoGrid, Google App Engine, Microsoft Azure, OpSource, and Rackspace have all been tested in practice by migrating Wera to the platforms. The migration processes are described in chapter 7.

5.1 Amazon Web Services

Amazon was originally a retail business selling books online and as the company grew, a global computing infrastructure was constructed. In 2006 Amazon decided to make part of the infrastructure available to customers as a cloud computing platform. Amazon Web Services, AWS, is a collection of services that constitute this cloud platform. [2]

5.1.1 Elastic Compute Cloud

The main product in the AWS suite is the Elastic Compute Cloud, EC2, which is an IaaS offering. EC2 offers the ability to start and stop instances on demand and only pay for the resources consumed. There are different instance "sizes" that correspond to hardware specifications. Amazon provides preconfigured Amazon Machine Images, AMIs, with a variety of operating systems including Windows Server, Open Solaris, and various Linux distributions. Partners and customers of Amazon are also free to create AMIs with software preinstalled; common examples of AMIs contain database, web hosting, and batch processing software. Amazon currently hosts AWS out of five regions, two in the US, one in Europe, and two in Asia. Each region in turn consists of availability zones which are geographically separated. Availability zones in the same region are connected through high speed networks and traffic between them is cheaper than regular internet traffic. The user can choose in what region to deploy a certain instance.

EC2 works in conjunction with other Amazon services which provides it with additional features. Elastic Load Balancing is one such service, providing EC2 with load balancing functionality. The load balancer manages a group of EC2 instances and detects unhealthy ones which are replaced automatically. The instances in a group can be in different availability zones to increase the robustness of the service. The load balancer offers the ability to define conditions for scaling the number of running instances up and down and the scaling is handled automatically. Use of the Elastic Load Balancer comes with an additional cost, based on the network traffic through load balancers and the number of hours each load balancer is used. [4]

5.1.2 CloudWatch

Amazon CloudWatch is a monitoring service that monitors EC2 instances, Elastic Load Balancers, Elastic Block Store volumes, and Relational Database Services instances. For Amazon EC2 instances, CloudWatch collects and reports metrics for CPU utilization, data transfer, and disk usage and activity for each Amazon EC2 instance. CloudWatch also automatically monitors Elastic Load Balancers for metrics such as request count and latency, Amazon EBS volumes for metrics such as read/write latency, and Amazon RDS DB instances for metrics such as available storage space. The basic CloudWatch functionality is included in EC2 and reports metrics every five minutes. Detailed monitoring comes with an extra fee and provides a one minute granularity. The detailed monitoring service also aggregates data for different AMI instance types. CloudWatch can be used to receive alarms when the system surpasses specified thresholds. The monitoring data is persisted for two weeks.

CloudWatch enables an EC2 feature called Auto Scaling. Based on CloudWatch metrics Auto Scaling can automatically start or terminate EC2 instances to accommodate the current load. The scaling can also be set to follow a predefined schedule. Auto Scaling can be used to keep the number of running instances constant by creating an Auto Scaling Group. The health of all instances will then be evaluated by Auto Scaling and non responsive instances will be replaced. CloudWatch has the ability to aggregate data from an Auto Scaling Group and show the user metrics for the group as a whole. [4]

5.1.3 Virtual Private Cloud

Amazon Virtual Private Cloud, VPC, is a service that lets a user integrate EC2 instances with the local network of the user. The local network and the VPC communicates via an

IPsec encrypted VPN connection and the EC2 instances are isolated from the rest of the Amazon cloud. The IP addresses of the instances are supplied by the customer and a VPC also supports subnets. All of the traffic between VPC instances and the public Internet is routed through the VPN connection. This makes it possible for a customer to use existing firewalls and intrusion detection systems to control the traffic to and from the VPC. Amazon VPC is currently in beta stage and is not covered by a SLA. [10]

5.1.4 Storage

The AWS platform offers a variety of options when it comes to storage solutions.

Elastic Block Store

The Amazon Elastic Block Store is one such solution, it is a block-level storage service meant to be used with EC2. One or more EBS volumes can be attached to an EC2 instance but the storage is persistent and not tied to the instance. A volume cannot be attached to multiple EC2 instances at the same time but can be reattached to different instances. The data is replicated within the same availability zone to make it durable. The capability to create snapshots of a volume and store it in another storage solution also exists. Volumes can be created and deleted, attached and detached with SOAP and REST APIs. EBS is designed for data that changes frequently and is therefore suited for databases or storage for a file system. [3]

EC2 ephemeral storage

The EC2 instances also comes with a local store volume, also called ephemeral drives. The storage is tied to an EC2 instance and cannot be reattached to a different instance, it is persistent over reboots of the EC2 instance but not if the instance is terminated or fails. Unlike EBS the storage is physically placed on the same machine as the running instance and the access performance is therefore high. Because of the lack of persistence and high access speed local instance store volumes are suitable for data that is temporary and constantly changing.

Simple Storage Service

Amazon Simple Storage Service, S3, is a distributed object store. It is designed to be very scalable, durable and highly available. S3 can handle objects from one byte to five terabytes in size and allows for concurrent read and write access from many clients or threads. The data is synchronously replicated over multiple devices and data centers to provide the high durability of S3; it is designed for 99.99999999 % durability and can manage concurrent data loss in two data centers. Just as EBS, S3 provides SOAP and REST APIs for retrieval of data. In S3 each object has a unique object key and can be accessed via a specific HTTP URL address. This makes it useful for storage of static web content. [7]

SimpleDB

Amazon SimpleDB is a non-relational data store. It is, like S3, designed to be scalable and highly available. To make the storage durable each data item is replicated to multiple locations within the selected region. SimpleDB stores data items comprised of a flexible number of name/value pairs and each domain of SimpleDB allows for ten gigabytes of

storage. Since SimpleDB is non-relational and schema-less it does not support SQL and cannot perform join queries. Instead the Select operation provided allows for queries that retrieve attributes based on criteria. SimpleDB can be managed with SOAP and REST APIs.

A common usage scenario is to store metadata for an object in SimpleDB and the actual data object in S3. [8]

Relational Database Service

Amazon also provides a relational data store in Amazon Relational Database Service, Amazon RDS. It is a fully functional MySQL database housed in the cloud. The database can be scaled in many ways; the capacity of the compute instance can be increased or decreased as well as the storage size and the number of read replicas. Read replicas use the built-in asynchronous replication of MySQL. Amazon RDS also offers a multi availability zone feature which sets up a standby replica in another availability zone. The replica is synchronously replicated and if the primary instance fails RDS will automatically transfer control to the replica. Backups can be performed in two ways with RDS. Daily backups can be activated to perform automatic backups that are stored up to eight days. The other option is user-initiated snapshots of the database that can be stored indefinitely and used to restore the database to a specific point in time.

Using a relational database manager other than MySQL can be achieved by using EC2 in combination with EBS. [6]

Simple Queue Service

Amazon Simple Queue Service, SQS, is a message queuing service. Messages are text based and can be up to 64 kilobytes in size. The storage is temporary but durable; data is replicated over multiple data centers. There is no limit for the number of queues or the number of messages per queue. The messages are kept in a queue until explicitly deleted or the expiration time expires. The expiration time can be set to a number between one hour and fourteen days. The queues can be accessed through SOAP and HTTP interfaces. SQS is suitable for producer-consumer scenarios, it enables different components to communicate asynchronously and allows for the number of communicating applications to grow or decrease when needed. [9]

5.2 Atlantic.Net

The company was established in 1994 and the headquarters are located in Orlando, Florida. Atlantic.Net is a hosting solutions provider and launched their cloud computing platform in December 2010. [1]

Atlantic.Net offers cloud servers, IaaS supporting Ubuntu, CentOS, Fedora, Debian, and Windows Server. For administration of the cloud servers, CPANEL/WHM can be added at an extra cost. Unfortunately there is not a lot of additional information about the platform available on Atlantic.Net's web site.

5.3 City Cloud

City Cloud is a product of the company City Network which was established in 2002 and is based in Karlskrona. City Network is mainly a traditional web host and introduced the

cloud computing platform in April 2010. [17]

City Cloud is the only Swedish company in the evaluation and the offering is an IaaS. The platform has been developed in collaboration with Dell, Cisco, and Enomaly, providing the servers, network, and virtualization platform respectively. The platform supports Ubuntu, Fedora, Open Solaris, Red Hat, and Windows Server. Custom images supplied by the customer is not currently supported however. For monitoring purposes, op5 Cloud Monitor can be added at an extra cost.

5.4 Flexiant Flexiscale

Flexiant is a software and services company based in Livingston, Scotland. Extility is a licensed product for hosting clouds on data centers developed by Flexiant and Flexiscale is a public cloud based on Extility. Flexiscale was released in October 2007 by XCalibre Communications, later acquired by Flexiant. The data center hosting Flexiscale is located in the South-East of England. [19]

Flexiscale is an IaaS platform offering CentOS, Debian, Ubuntu, and Windows Server images to run on the service. Customer created images are also supported. Each customer gets a private VLAN and the storage is virtualized and is kept in a separate Storage Array Network rather than in the physical hosting server. The separation of storage and the hosting server gives the capability to turn off a server and only pay for the storage for a period of time. The platform supports snapshots of existing servers enabling backup or starting of multiple servers with the same configuration. Flexiscale monitors physical servers and restarts instances on crashed servers automatically on other servers within fifteen minutes.

5.5 GoGrid

The GoGrid platform was launched in March 2008 and the company is solely a cloud hosting company. It is based in Silicon Valley, California and has two data centers in the US, and another one planned in Europe in the summer of 2011. [22]

GoGrid offer an IaaS platform supporting CentOS, Red Hat Enterprise Linux, and Windows Server. My GoGrid Server Image, MyGSI, is a feature allowing users to save images of a server with the software and configurations of their choice to quickly add new servers that are already configured correctly. The GoGrid Exchange lets GoGrid partners share or sell images with their software preinstalled, giving GoGrid users easy access to different applications. Vertical RAM Scaling is a GoGrid feature enabling on demand scaling of instances when the load on an application changes. The amount of RAM on a server can be scaled up or down but not lower than the initial configuration of the server. GoGrid servers can be managed through the web portal or with the REST-like API provided; features include choosing which data center to deploy a server in. The offering includes a hardware load balancer that distributes calls to servers based on their current load or alternatively, in a Round Robin fashion.

GoGrid also offers Cloud Storage, a file storage service. Cloud Storage can be accessed by servers with SCP, FTP, SAMBA/CIFS or RSYNC. A content delivery network (CDN) with points of presence in 18 locations in the US, Europe, Asia, and Australia is also available as an add-on feature. The CDN supports live video streaming, a large object download service, and security through SSL among other things. An additional add-on feature is a hardware firewall supporting VPN access to servers.

5.6 Google App Engine

Google was established in 1998, at the start with a focus on its search engine. After experiencing a rapid growth Google has widened its scope developing many services ranging from e-mail to map services. App Engine, Google's cloud computing platform was released for developers in April 2008. Many of the services are available at no cost for the users. [23]

5.6.1 The Platform

Google App Engine offers the ability to create web applications that will be executed on Google infrastructure. Google App Engine is a PaaS offering with software development kits and runtime environments for Java and Python. The Java runtime environment of Google lets the user create application with not only Java but also languages such as JavaScript and Ruby that produces bytecode readable to the Java virtual machine.

Running an application on the Google infrastructure means automatic scalability and load balancing for the application. Each application has a queue for incoming request that is monitored by App Engine. When the queue becomes too long more instances of the application will be started to share the load. The same applies when the load on the application decreases, the number of instances will then be reduced. If an application is not used at all for a period of time it will be turned off until a new request arrives. An always-on feature can be bought to prevent an application from being shut down at \$0.3/day.

5.6.2 The Datastore

Another piece of the Google infrastructure that is used by App Engine applications is the datastore which is based on Google's BigTable and MegaStore. The datastore is a distributed data storage service with support for transactions. It is a non-relational schemaless datastore that contains entities which have a kind and a set of properties instead of tables and rows. The query language is therefore not SQL but instead a querying language called GQL which has a similar syntax can be used. Since the datastore is designed with scalability as a priority, there are some restrictions to the queries. There are no join operations and inequality filters can only contain one property per query. Google App Engine comes with two options of datastore, the Master/Slave Datastore and the High Replication Datastore. The two options offer the same API for the application developer but differ in the back-end.

The Master/Slave Datastore uses asynchronous replication and is the default option. The disadvantage of this option is that the master is a single point of failure and planned, or unplanned, downtime of a data center can make writing to the datastore impossible. The other option is the High Replication Datastore which is more robust and resilient to failure, one data center becoming unavailable will not render the datastore unusable. The data is distributed over datacenters using a solution based on the Paxos algorithm which is an algorithm for reaching consensus between replicas in a distributed system [33]. The increased robustness comes with some disadvantages though, the latency for altering the datastore is slightly increased and the high replication consumes more resources, approximately three times the storage and three times clock cycles for altering the datastore. Because of the increased resource consumption and subsequently increased cost of the High Replication Data Store, Google only recommend it for "mission critical applications".

5.6.3 Billing

The App Engine pricing model is different from the other evaluated platforms in that the use is free up to a certain amount of resources consumed each day. An application exceeding its quota will stop functioning until the next day or until the billing feature is enabled. The billing feature lets the user set a budget for each day. The user will then be billed for any resources consumed exceeding the free quota. There are two types of limits, fixed limits that cannot be overdrawn and billable limits where more money buys a customer more capacity. This quota system has detailed limits for the different parts of the system and the operations performed by each system. The whole comprehensive quota specification is too long to present here but can be found at [24]. However, the fixed limits are in general quite generous, some examples of these limits are: 558 GB data sent to memcache API, 417 million queries to datastore, and 43 million requests to the application. Examples of the billable limits are 1GB in- and outgoing bandwidth per day, 6.5 CPU hours per day, and 1/0.5 persistent storage GB for the data store and high replication storage.

5.6.4 App Engine for Business

When App Engine was first launched in 2008 one of its main objectives was to make it easy for developers to create applications and making them available to the world without having to invest in hardware or managing their own software stack for the application. With restrictions like a maximum of ten applications per user and the lack of a service level agreement App Engine has been less suited for businesses. To meet the demands of business customers Google are planning to release Google App Engine for Business. New features include a central administration console for all the applications of a company, hosting of SQL databases, and a service level agreement with promises of 99.9 % availability. There is a new pricing model for intranet applications at eight dollars per month and user, but with a maximum of a thousand dollars, for each application. The pricing model for public applications is not yet published nor is the details of the service level agreement. Google has also entered a partnership with VMware to increase the portability of applications developed for App Engine. With the development tools made available applications will be able to run on App Engine, a private cloud with VMware software or another cloud infrastructure such as Amazon EC2. All the features of Google App Engine for Business are not available yet but will be so in the second quarter of 2011 according to Google's roadmap.

5.7 Joyent

The company is a cloud computing services and software company, Joyent has been offering its services since 2004. The headquarters are situated in San Francisco, California and data centers hosting the Joyent cloud are located in the US, Europe, and Asia. [29] [30]

Joyent has developed their own cloud computing platform, based on the operating system Open Solaris they have created Joyent Smart OS that replaces the hypervisor and traditional operating systems used in other IaaS platforms. This platform is sold by Joyent to customers wanting to build a cloud of their own and is also used to provide a public cloud managed by Joyent. Joyent offers two kinds of products on their public cloud, SmartMachines and Virtual Machines. SmartMachines uses the Joyent Smart OS and comes with Apache, Python, Ruby on Rails, Java, and SVN preinstalled. There are also SmartMachines specialized for MySQL, Riak (a scalable open source key/value store database), and Zeus (a load balancer) available on the Joyent cloud. SmartMachines share a hardware resource pool and are capable of CPU

bursting when load increases. Virtual machines run CentOS, Debian, Ubuntu or Windows Server operating systems. Joyent does, in contrast to most other vendors charge by the month instead of by the hour for used instances.

5.8 Microsoft Azure

Microsoft was founded in 1975 and is headquartered in Redmond, Washington. Microsoft launched Windows Azure in October 2008. [20]

Azure is Microsoft's take on cloud computing. Windows Azure is a PaaS offering where developers can create applications using .NET, Java, Ruby, or PHP. In addition, Windows Azure AppFabric and SQL Azure and Windows Azure Marketplace compliment Windows Azure to constitute the whole Microsoft cloud platform.

5.8.1 Windows Azure

Windows Azure in turn is made up of different components, namely, compute, storage, connect, fabric controller, and CDN.

Compute

Compute is the component that runs the applications. Applications can be created using three kinds of roles: web roles, worker roles and VM roles. Applications can have just one role or several and each role can be run in one or more instances. Web roles are used for web based applications and are invoked by web requests. Worker roles are intended to run background jobs and do not interact with the user directly. VM roles are a bit different; they run Windows Server images which is more of an IaaS concept. The compute component also contains a load balancer distributing jobs between the available instances. In order for the application to be scalable, which is a big part of the incentive for creating an Azure application, the instances has to be stateless since there is no mechanism to ensure a client is handled by the same instance over many requests. Client specific information has to be stored in Azure storage or by other means be made available to all of the instances.

Storage

The storage component offers three kinds of storage: blob, table and queue. Blobs are held in containers which can make up a hierarchy. They can store large data objects, up to a terabyte, and contain metadata about the data object. To optimize transmission of these potentially large blobs, they can be divided into blocks allowing retransmission of separate blocks in case of a failure. For data that require a more structured storage tables can be used. These are not relational tables, and consequently SQL is not used to access the data. Instead the data is accessed through a REST API and the data objects, or entities, has properties of types as int, string, and bool. The use of a NoSQL storage model like this enables the data storage to scale and the data to be partitioned over several servers. The third part of storage, queue, is not intended for persistent storage in the same sense as blob and table. The queue structure provides a natural way for different roles to communicate. A web role that receives a request which demands heavy computations can hand it off to a worker role through a queue and the worker role can then hand it back through a different queue when the work is complete.

The data stored in Azure storage is replicated three times to mitigate any data loss.

CDN

The CDN, or content delivery network, is aimed at improving performance for data that is accessed frequently from places all around the world. By storing local copies at different sites the access speed can be improved.

Connect

The connect component enables an Azure application to connect to another computer at the IP level rather than HTTP, HTTPS and TCP which are the normal protocols for connection to computers outside of the cloud for applications on Azure. This can be useful if an application is to connect to a database on a local machine running SQL Server. This solution requires software on the local machine but enables the cloud application to connect as if the two machines were on the same IP network. The connect component can also be used to join a cloud application to a local Active Directory to enable single sign-on and use the Active Directory accounts for access control.

5.8.2 SQL Azure

The storage model of Azure storage is non-relational because of the ability to scale more easily. However, many organizations use relational databases and Microsoft provides the ability to move them to the cloud with SQL Azure. Databases on SQL Azure can be used by applications ran on Azure, a local machine, or a mobile device. SQL Azure works much in the same way as SQL Server, some features are still missing but will be made available in future versions[15].

The database can be accessed with either a protocol called Tabular Data Stream, the same protocol used to connect to a local SQL Server database, or Open Data Protocol. Each SQL Azure account can have one or more logical servers and each server one or more databases which can be up to 50 gigabytes in size.

Just as in Azure storage, data stored in SQL Azure gets replicated three times and the database upholds the property of consistency.

SQL Azure also provides reporting, SQL Azure Reporting, based on SQL Server Reporting Services. Using this tool reports can be published to a SQL Azure Reporting portal and from there be integrated into an application.

SQL Azure Data Sync allows a SQL Azure database to be synchronized with another database, a local SQL Server database or another SQL Azure database. The first option can be useful if an organization wants to ensure access to the database even if the network connectivity is lost. The other option can be used to host the same database in different data centers around the world to increase performance for clients in different regions. The synchronization can be started manually or by a scheduling service which can synchronize a pair of databases at a certain interval.

5.8.3 Windows Azure AppFabric

AppFabric provides three Azure based infrastructure services: Service Bus, Access Control, and Caching. These functions address common challenges in distributed applications and can be used by both cloud based and local applications. The number of functions included in AppFabric is going to increase in the future according to Microsoft's plans.

Service Bus

Service bus is aimed at making a web service built with Windows Communication Foundation accessible to applications outside of the cloud or local network. This works in the following way[15]:

1. The web service registers its endpoint at a registry in Service Bus.
2. Service Bus exposes the endpoint and assigns an URI to it.
3. A client wishing to access the service contacts the Service Bus providing the URI and receiving a service document.
4. With the service document, the client invokes operations on the Service Bus.
5. Service Bus in turn invokes the operation on the service.

By these means, the registry provides a way for the client to find the service end point. Other problems solved by the service bus is that the web service may not have an external IP address if it resides inside a network with NAT and that the requests to the web service may have to get through a firewall. In step one above, the TCP connection to the Service Bus opened by the web service is kept open so that subsequent invocations to the web service can use the same connection. This means that they will be routed to the web service and the firewall will not interfere since the connection was initiated from within the network.

The Service Bus can also work as a layer of security since the IP address of the web service is hidden making the application anonymous.

Access Control

In a distributed system, access control is an important aspect. Claims based identity is a model to handle this, users are issued tokens by identity providers and the tokens are then sent to the application to authenticate the user. There are many identity providers available on the internet such as Windows Live ID, Google, Yahoo and Facebook. Azure Access control lets an application accept tokens issued by different identity providers. This works in the following way[15]:

1. The user tries to access the application and the application redirects the call to the corresponding identity provider.
2. The identity provider authenticates the user and returns a claims token.
3. The token is the sent to Azure Access Control.
4. Access Control validates the token and creates a new one containing claims according to rules for the specific application, the new token is sent back to the user.
5. The Access Control token is sent to the application and is validated giving the user access to the application.

In this way, support for many identity providers can be added to an application without the developer having to worry about different format of tokens.

Caching

Caching is a common way to increase performance and Azure provides a distributed cache to potentially speed up applications. Data is not automatically cached; this has to be done explicitly by using the Caching API. An exception is ASP.NET applications which can be configured to cache Session object data without editing the code of the application. The cache is distributed in the way that an item not found in the local cache will be searched for in the caches of other instances running the same application.

5.8.4 Windows Azure Marketplace

The Marketplace will consist of two components, the AppMarket and the DataMarket where applications and data sets will be purchasable. The DataMarket will be made available first. The data available in the Marketplace can be viewed by users through the Service Explorer and also be accessed by applications by REST and the Open Data Protocol. After a data set has been purchased it can be stored at Azure Storage, SQL Azure or a local database. Microsoft will review the quality of data providers and to begin with only make available data from the top five providers in a particular industry.

5.9 OpSource

OpSource is a managed hosting company that was founded in 2002 with headquarters in Santa Clara, California. In August 2009 the OpSource Cloud was released and in May 2010 the Cloud Files service was added to the offering. OpSource delivers its cloud from facilities in the Sunnyvale, California; Ashburn, Virginia; London, UK; and Paris, France.

The OpSource Cloud is an IaaS offering. In addition to Cloud Servers, Cloud Files and Cloud Networks are also parts of the platform. Cloud Servers support Windows Server, Ubuntu, CentOS and Red Hat Enterprise Linux. Cloud Servers and Cloud Files can both be accessed through a REST-based API. The OpSource Cloud Network refers to the physical network structure of OpSource. All cloud customers receive a VLAN separating their cloud servers from other OpSource servers, more VLANs can be created for an additional fee. Cloud Networks offer features such as VPN connection to servers, load balancing, and layer two multicast. Cloud Files allows for creation of multiple storage accounts with separate administrative passwords per customer. Each account can store up to ten terabytes of data and there is no specific file size limit. Data Stored in Cloud files is encrypted with 256 bit AES.

5.10 Rackspace

Rackspace was established in 1998 and is based in San Antonio, Texas. It is a hosting company that has now branched into the cloud computing business, the Rackspace Cloud, previously known as Mosso, was released in its current form in March 2009. [13]

The Rackspace Cloud consists of two components, Cloud Files and Cloud Servers, as the names suggests Cloud Files is a storage service and Cloud Servers IaaS.

5.10.1 Cloud Files

Cloud Files has a file size limit at 5 GB and files can be uploaded through the online control panel, the Cloud Files API, or via third party software. Files are replicated to

three different data center zones with separate power and network connections. Rackspace provides a content delivery network in collaboration with Akamai that is integrated with Cloud Files. The CDN has locations all over the world and works automatically with no programming required. When a user downloads a file, the file is saved at an edge server close to the user reducing delivery time for the next download in the region.

5.10.2 Cloud Servers

Cloud Servers support Ubuntu, Debian, Gentoo, CentOS, Fedora, Arch, Red Hat Enterprise Linux, and Windows Server. Customer created images are not supported. The platform is based on Xen hypervisor for Linux and XenServer for Windows. Each virtual server gains access to CPU cores and cycles according to its size. If resources are available, virtual servers can use CPU burst to temporarily increase the computing power. RAID-10 is used for the storage of Cloud Servers to preserve the data in case of a host failure. Snapshots of images can be created on demand or according to a schedule and used to backup the servers.

5.11 Salesforce Force.com

Salesforce started as a CRM software provider and was a pioneer in the SaaS field. Today their SaaS suite consists of Sales Cloud, Service Cloud and Chatter. These applications are as suggested by their names a sales application, an application for customer service and an application to support collaboration within an organization.

Salesforce also offers a PaaS suite, Force.com, consisting of Appforce, Siteforce, VMForce, Heroku and ISVforce. The target for Appforce is business applications which are developed with the Salesforce proprietary point-click-language APEX. The 80 % clicking and 20 % coding model is supposed to speed up the development time and thereby reducing costs. Applications created with Appforce can be integrated in the SaaS applications. Siteforce is aimed at making websites easily, as with Appforce, the coding required is minimal. The VMForce platform is a result of a partnership with VMware and enables java applications to run on the Salesforce infrastructure. Heroku is a PaaS for Ruby recently acquired by Salesforce. ISVforce (independent software vendors) offers software vendors the opportunity to become partners to Salesforce, developing their application on the Salesforce infrastructure. The partner vendors can then make the applications available as SaaS offerings on Appexchange, an application market place hosted by Salesforce.

Chapter 6

App Engine vs. Azure

Microsoft and Google are offering Platform-as-a-Service (PaaS) level solutions with general applicability and are the greatest in this section by far. Therefore, a special focus will be devoted to comparing the two.

Neither of the platforms has reached a stage of maturity. Many of the Azure services offered are still in the beta stage and not covered by an SLA and there are still many features in planning. As for App Engine, the whole platform is still in Preview status and therefore has no SLA or promise of a certain rate of uptime. Because of this some businesses are reluctant to invest too heavily in the platform and for now, this tilts the advantage in Azure's favor in some regards. However, App Engine will become an official Google product later this year and will then be adding features such as a SLA and operational support for the platform.

6.1 Level of Virtualization

Although Azure and App Engine can both be seen as PaaS offerings they do differ in the level of virtualization. In [18] Amazon Web Services are placed at one end in the spectrum, App Engine in the other, and Azure in between. The difference in the level of virtualization is noticeable in many ways.

In contrast to most other platforms, App Engine has no concept of instances from the user's perspective. A user cannot provision an instance or request a number of instances for an application, the provisioning and scaling is performed internally by App Engine. Azure has automatic scaling, just like App Engine, but the user has control over the maximum number of instances running an application. Also, when using the model of instances provisioned by the user, all applications will need at least one instance to be running (or two if the application uses worker roles) even when there is no activity. An application on App Engine with no activity for some period of time will shut down. When a request arrives a new version of the application will be started to handle it. This is only an issue for applications with little traffic. For such applications Azure has the disadvantage of instances incurring charges while idle and App Engine has the disadvantage of responses taking a bit longer than usual if the application was turned off.

Azure also provides the VM-roles allowing for the user to upload a virtual Windows Server machine to run on the platform. There is no corresponding feature on App Engine and this offering that is basically at the infrastructure level further establishes Azure as a lower level offering in the PaaS space.

The geographical placement of the data and application is another thing that is abstracted away in App Engine. There is no way to know or control where the machine hosting an application is located geographically. Azure is run on two datacenters in North America, two in Europe, and two in Asia and the user can choose in which region the application should be hosted. The geographical location of the data may be important for some applications due to legal issues.

6.2 Storage

The storage capabilities of the platforms may seem dissimilar at first glance but there are in fact many similarities. The datastore of App Engine and the table storage of Azure are both distributed storage solutions lacking support for the conventional relational query set. For storage of larger files App Engine provides a service called Blobstore and the Azure alternative is the Blob Service of Azure Storage. There is one difference to note regarding the storage of blobs which is the maximum size of the blobs. The limit is 2 GB on App Engine and 1 TB on Azure. Another of Azure's storage services is the Queue Service for communication between web- and worker roles. App Engine provides Task Queues for the same purpose. However, differences do exist; for example regarding the file system access and relational databases. App Engine applications cannot write to the local file system where as this is possible on Azure if instance storage is requested by the application. One part of the Azure platform is SQL Azure which provides a relational database hosted on the Azure infrastructure. There is currently no such alternative on App Engine but some sort of support for this will be added during 2011¹.

6.3 Support for Application Types

Azure has been targeted at general purpose applications from the beginning offering both web- and worker roles for serving web pages as well as performing batch computations. App Engine was at the start focused at traditional web applications, with the state saved in the datastore and the computation nodes kept completely stateless. The applications were also expected to be request-reply based. Because of this App Engine were deemed "not suitable for general-purpose computing" in [18]. More examples of criticized restrictions of App Engine are a 30 second timeout for responding to each request and that applications cannot spawn new threads. However, the level of support for general-purpose computing on App Engine are changing, very recently (May 2011) the App Engine team announced *backends*. Backends have roughly the same purpose as the worker roles of Azure. They have higher memory and CPU limits, persistent state across requests, and no request deadlines making the platforms more alike in terms of which types of applications that can be hosted. It may be too soon to tell how well the new features of App Engine will function but the support for various application types that have previously been an advantage for Azure may be going away as App Engine launches features to broaden the scope for the platform.

6.4 Software Development

When developing for the platforms both come with SDKs and emulators for simulating the cloud environment locally. Both of the solutions work well but neither have the complete

¹<http://googleappengine.blogspot.com/2011/05/year-ahead-for-google-app-engine.html>

feature set of the actual platform. Both also supply the capability to upload a version of an application without taking down the version facing the public. Azure has a staging and a production stage and App Engine supports different versions of the same application. One notable difference discovered during the migration phase of this study is that deploying an application to Azure takes close to ten minutes while doing the same to App Engine takes a few seconds.

6.5 Related Studies

Azure and App Engine has been compared before, in [38] the author compares the platforms by developing the same application on both platforms and concludes that App Engine provides an easier learning curve but that Azure is a good choice for existing .NET developers. In [39] the platforms are compared in several categories. Azure is crowned as winner in language support, application types, and customized solutions. App Engine wins in Development/ease of migration and cost of ownership while the platforms are tied in scalability and storage. However, comparisons like these quickly get outdated since the platforms are constantly changing and new features are being added.

Chapter 7

Migrations

After the evaluation of cloud platforms Wera was to be migrated to the most promising and interesting cloud platforms. Initially the plan was to perform two or three migrations but the time it took to migrate the application to an Infrastructure-as-a-Service (IaaS) platform was shorter than expected. Therefore, migrations were performed to all infrastructure level platforms except for Joyent and Atlantic.Net. Joyent was excluded because it is mainly geared towards Smart Machines which do not run Windows, also they do not employ the same fine granularity in the billing model as the alternatives. Atlantic.Net was excluded since it was deemed as the weakest of the candidates and lacked any distinguishing feature. Here follows a description of the migration processes and the experience gained from migrating Wera to the infrastructure level platforms and the Platform-as-a-Service (PaaS) alternatives Microsoft Azure and Google App Engine.

7.1 Infrastructure-as-a-Service Platforms

Wera has been successfully migrated to the following IaaS platforms:

- Amazon EC2
- City Cloud
- Flexiant Flexiscale
- GoGrid
- OpSource Cloud
- Rackspace Cloud Servers.

The process of a simple migration scenario to an IaaS platform looked like this:

1. Create an account on the platform.
2. Create a server from a Windows Server template.
3. Install any missing software e.g. SQL Server (Express), .NET Framework.
4. Add the role "web" to the server.

5. Restore the database of Wera on the server.
6. Move the application to the server and configure connection settings.
7. Configure the Internet Information Services web server.

Once these steps are done the application is working normally in the cloud. For a production installation there are additional steps to be taken. Some strategy needs to be put in place to replicate or back up the database. The IP address given to the server needs to be mapped to the desired URL for the application. In case the server crashes there needs to be a snapshot present to start a new one to replace it and the instantiation of the snapshot should preferably be automated. Firewalls, on the server or supplied by the platform, also need to be configured.

When leveraging cloud computing platforms to run applications the idea is that the user should not notice the difference. The same applies for the developer or system administrator who is managing an instance on a cloud platform. The instance essentially behaves as any Windows server, hosted locally or on any infrastructure platform. The most important difference between IaaS platforms that can be observed from within a server is the performance of the server, network, and storage. Since this evaluation does not focus on the performance aspect, the interesting differences to be noted during the migrations were the quality of the administration interface and the service in general as well as the functionality of the platform regarding cloning of instances, storage options, backup solutions, etc.

7.1.1 Registration Process

All of the platforms allow users to create accounts via the Internet and start using the platform directly.¹ The only information required is the user's personal data and a valid credit card. Amazon also requires a working phone number which is verified by an automatic call where the user has to enter a PIN that appears in the browser. Microsoft requires a Windows Live ID² to sign up for Azure and the account will be used for logging into the cloud platform administrative interface. After signing up for Rackspace cloud a representative of theirs will call the user and confirm the credit information and answer any questions the user might have. Before an account at GoGrid is activated the user receives an e-mail from an "Account Development Manager" with inquiries regarding topics like the current hosting topology, what problems the cloud platform is intended to solve, and the long term plans for using the platform. GoGrid also offers a "on-boarding session", an online meeting where a GoGrid customer service representative gives an introduction to the administrative interface. Since GoGrid are based in California the time difference may be a problem for Swedish customers.

7.1.2 Snapshots

The ability to take snapshots of instances with certain software stacks installed and then start new instances that are already preconfigured is vital for basically any usage of an IaaS platform. Without this functionality, the on-demand scalability would not be as much of an advantage for handling peaks in traffic since installing all required software from scratch might take a long time and not be feasible in practice.

GoGrid, OpSource, and Rackspace all allow for snapshots to be stored in the corresponding cloud files storage solutions of the platforms. Flexiant offers a similar functionality where

¹GoGrid requires some additional information by e-mail before the account is activated

²www.passport.net

snapshots can be taken and a clone of the server disk will be saved. City Cloud does only support cloning of servers, there is no snapshot functionality where a particular image can be saved. However, servers that are shut down can be cloned and only incur charges for the storage. Therefore this is essentially the same as a saved snapshot but a less elegant solution. One difference to note here is that different methods can be used to store the snapshots. The more simple approach is an exact clone of the disk which consequently is the same size as the disk; this approach is employed by OpSource, Flexiant, and City Cloud. The more advanced option is *thin provisioning* where only the actual files are backed up making a snapshot of a disk with unused space smaller than exact clones; Rackspace and GoGrid snapshots work this way.

Amazon provides a greater variety in choices for snapshots. The concept of an Amazon Machine Image (AMI), which is a pre-configured image for instantiating servers, is used in their solution. Many AMIs with different operating systems and configurations are supplied by Amazon and even more options are made available by third parties offering both free and "paid for" AMIs. Any user can also create their own AMIs and chose to share them with the community or not. There are two different solutions for creating AMIs; EBS-backed AMIs and S3-backed AMIs. These two options differ in many ways. The AMI itself is stored in either EBS or S3 as the names imply. The root device of an EBS-backed instance is an EBS volume, which makes the storage persistent on instance failures and allows for the instance to be stopped. S3-backed instances uses ephemeral instance storage for the root device, the data is therefore only persistent for the life of the instance, the data is lost if the instance fails or is terminated and stopping the instance is not possible. Since EBS-backed instances can be stopped it is possible to modify the instance type, RAM, disk, and user data of an instance which is not possible with S3-backed instances. EBS-backed instances are faster to boot than S3-backed instances and also have a higher size limit, 1 TB compared to 10 GB. The cost of the two solutions also differ, the EBS choice uses one EBS volume for each instance and one for each AMI while the S3 alternative means that the AMIs are stored in S3 and the instance storage is free. EBS storage is also more expensive than S3 but the difference in cost can be mitigated by the fact that when backing up AMIs only the changes are saved for EBS-backed AMIs while the whole AMI is saved again in S3, potentially wasting a lot of space. The features of the two alternatives can also be reviewed in table 7.1.

	EBS-Backed	S3-Backed
Max instance size	1 TB	10 GB
Root Device Location	EBS volume	Instance storage
Data persistence	Persists instance failure, can persist instance termination	Persists only for lifetime of an instance
Upgrading	Instance type, RAM, disk, and user data can be changed while instance is stopped	Instance attributes are fixed for the life of an instance
Charges	Instance use, EBS volume, and EBS snapshot	Instance usage and S3 charges for snapshot
AMI creation	Single call/command	Requires installation and use of AMI tools
Stopped state	Can be stopped and data persisted on EBS volume	Cannot be stopped without being terminated

Table 7.1: The differences between EBS- and S3-backed AMIs

7.1.3 Load Balancing

On GoGrid and OpSource a more advanced set-up than using a single server for the whole application was constructed. Figure 7.1 shows the set-up which consists of a load balancer, a firewall, two web servers, and a database. Both of the platforms supply load balancers at no additional cost and both has two usage modes, sending traffic to the least busy server or directing traffic in a round robin fashion. The database used was the SQL Azure database also used for the Windows Azure migration. The process of setting up a group of load balanced servers was fairly easily done on both platforms although easier on GoGrid where the process was more intuitive and consisted of fewer steps. The OpSource cloud has the concept of a server farm which is a group of servers that are reachable through a load balancer. To create a server farm each server that will belong to the group needs to have a "real server" associated with in it the load balancer. Then a server farm is created with one of the real servers set as the initial server and the remaining real servers are then added to the server farm. Finally, a Virtual IP is created that targets the server farm and the setup is complete. The GoGrid administrative interface is more straightforward in this regard, a grid view shows all instantiated servers and load balancers as icons giving a clear overview of the network. To set up a group of load balanced servers a load balancer is created and the servers are then added to it. Rackspace Cloud and Amazon EC2 also supply the capabilities to

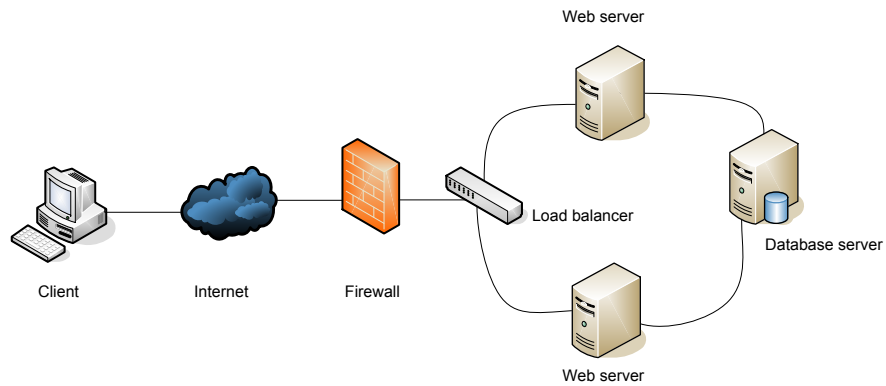


Figure 7.1: Set-up with multiple web servers.

create load balanced groups of servers in similar ways. On Rackspace and Amazon though, each load balancer and the traffic through them comes with an additional cost. If load balancing is required for a system running on Flexiant Flexiscale or City Cloud it has to be solved using special software on a regular instance since there is no such functionality in the platforms.

7.2 Microsoft Azure

The next platform to migrate Wera to was Windows Azure. When running an application on Azure there is the choice of web roles, worker roles, and VM roles. Since a VM role resembles the IaaS model this alternative was not investigated further, the PaaS aspects of Azure were more interesting to review. A worker role was not necessary for Wera since no

heavy computations are performed, so a web role would suffice in this case. However, before the application was migrated the database was migrated to SQL Azure.

7.2.1 Migrating to SQL Azure

SQL Azure is a relational database provided as a utility. It works like SQL Server in many ways but there are some differences as well. The duties and opportunities of the database administrator to configure the physical hardware that the database runs on are removed when using SQL Azure. However, the logical administration of schemas, indexes, queries, and the managing of roles, users, and more still has to be performed by a database administrator. The physical administration possibilities are removed in order for SQL Azure to be able to automatically replicate data, perform load balancing, and handle potential failures with transparent fail-over to healthy replicas.

SQL Azure does, like SQL Server, provide a Tabular Data Stream interface for communication with the database making applications designed for SQL Server databases compatible with SQL Azure databases as well. It is also possible to administrate a SQL Azure database from a local installation of SQL Server Management Studio³ or from the command line utility SQLCMD. There is also a simple web based interface for managing databases in SQL Azure where new queries, views, and stored procedures can be created. The web interface also allows for firewall rules to be set controlling which IP addresses are allowed to access databases.

An account for Azure was created but when the data migration was to be performed a problem was discovered. Since a Live ID account is tied to each Azure account, an administrator using his or her personal Live ID may not want to share this account with the co-administrators, there is a need for some sort of user management. This is provided by Azure; co-administrators can be created and given the rights to administrate certain projects. However, this feature does not yet cover SQL Azure, only the account owner has access to the storage subscriptions and databases.

The migration of a SQL Server database to SQL Azure can be performed in some different ways. A Transact-SQL script for creating the database can be created using SQL Server Management Studio. The script may have to be modified in order to support SQL Azure though, since all the features of SQL Server are not supported. Once the script has been modified to be supported by Azure, the script could be executed against the database generating the tables and stored procedures. The data migration can then be performed using SQL Server Integration Services which can be used to get data back out of the cloud database as well.

An alternative to this is a popular tool called SQL Azure Migration Wizard. It is created by Microsoft employees but not officially supported by Microsoft, the tool is made available at Codeplex⁴. The tool can be used to analyze a database; it will then attempt to find any parts of the database that is not compatible with SQL Azure. A migration script can be generated and some of the unsupported features can be automatically modified by the tool, others have to be managed by hand. Finally, the migration of the tables, stored procedures, and data is handled when the connection information for the database is supplied.

The Microsoft Sync Framework is a third option for migrating a database. Aside from migrating data the framework can be used for synchronization between cloud databases and local databases or between SQL Azure databases in different data centers.

³This requires the latest version, 2008 R2, of SQL Server (Express).

⁴<http://sqlazuremw.codeplex.com/>

The first two approaches have been employed with success for migrating the database of Wera. Limitations with SQL Azure that were encountered in Wera's case were that a stored procedure contained an unsupported function and that SQL Azure requires clustered indexes on each table. These issues could be resolved with relative ease but for databases with more unsupported features a migration could be more complex.

7.2.2 Migrating to Windows Azure

Microsoft provides a Windows Azure SDK for developing applications for Azure; they also provide Windows Azure Tools for Microsoft Visual Studio that adapts Visual Studio for development of cloud applications. One feature of the SDK is a compute and a storage emulator for testing the application locally during development. This makes debugging much easier; on the actual platform debugging has to be done using log files. It also speeds up the testing since deploying the application to Azure frequently is time consuming. However, the emulator environment is not exactly the same as the cloud platform so testing will need to be done in the cloud as well.

The Azure tool for Visual Basic helps in many ways when migrating an application. The tool makes it possible to convert an ASP.NET application to a web role. This is done by creating a new Azure project and adding the existing application as a web role. This might work right away for some applications but the conceptual differences for running an application in the cloud may demand some changes to be made in the application. This was the case with Wera. Once the application is ready for Azure it can be deployed to Azure from within Visual Studio.

Configuring the application

A web role has two configuration XML files, named `ServiceDefinition.csdef` and `ServiceConfiguration.cscfg`. The service definition file contains definitions for the roles of the application like what instance size to be used, local storage for the instances, connection strings to databases, virtual applications or directories, environment variables, and more. The configuration file states the number of instances that are to be deployed, the values for configuration parameters that have been defined in the definitions file, and thumbprints for management certificates used by the application.

Wera has a modular design with many web services handling different parts of the application. This turned out to be somewhat of a problem for the migration. When the application was "converted" to a web role the referenced web services were not included in the result. The error messages produced by Azure was of no assistance but eventually it turned out that a solution to the problem was to add the web services to the Azure application as virtual applications in the service definition file. This made the web services available to the web role although at a different path than in the original application which required additional modifications of the application.

Differences from running locally

The original application used the local file system for two purposes, storing a log file and attachment files that are referenced from the database. There are two problems with this. A role on Azure does not have access to the local file system on the machine where it is running. Instead, instance storage has to be defined in the definition file, the amount of storage that can be requested depend on the instance size. Instance storage can be used like a normal file system once a handle to it has been retrieved with a method call. Once

instance storage is in place, the second problem is that it is not persistent in the same way as the local file system on an on-premise server. Instance storage is not replicated so any data kept there will be lost if an instance is terminated or fails. It is also not accessible to any other instance of the application. This makes it a poor fit for log files that might be needed in just the situation that an instance fails, and an even worse alternative for attachment files that should be stored permanently and be available to every instance.

A better alternative would be Azure Storage which as described previously offer blob, table, and queue storage. The blob storage is the most suited for large files and even though attachment files are restricted to be moderately small the blob storage was used for attachment files and the table storage was used for the log files to try out the techniques. The SDK for Azure contains classes that represent storage accounts, blobs, and containers that use the REST API in the background and the operations available in the API can be performed through method calls.

The blob storage is easy to set up and use. With only a few lines of code blobs are created and stored on Azure. There exist two variants of blobs, block blobs and page blobs. Block blobs are intended for storing files and was therefore chosen for the attachment files. Page blobs can be used for mounting virtual drives on instance roles or for other files with range-based updates. The table storage requires a slightly more complicated set-up. A class has to be created to model the schema of the table, the class should include the attributes `Timestamp`, `PartitionKey`, and `RowKey` which are required for every entity stored in the table storage. Then, a class deriving from `TableServiceContext` has to be created with methods for accessing the table. Though, when the classes needed are in place, new log entries can be added with ease.

7.3 Migrating to Google App Engine

Since App Engine currently supports applications written in Python and Java, converting a .NET application to run on the platform is not possible, it has to be completely re-written. Therefore a more simple case management application has been created as a "proof of concept" for the platform. The application has been developed using Java and Gaelyk. Gaelyk is a Groovy toolkit for creating applications for App Engine[21]. Groovy is a programming language which can also be used as a scripting language, it produces byte code that can be run on the Java Virtual Machine [25].

Google provides an App Engine SDK for developing applications and there is also a plugin for the Eclipse IDE. The plugin simplifies the process of creating projects and deploying applications to the platform. Just as for Azure there is an emulator to simulate the App Engine environment when running applications locally. The emulator is very useful during development but testing needs to be done on the platform as well since all functionality is not supported, for example the users service.

App Engine uses the Java Servlet Standard for web applications[28]. This means that applications are composed of Java Server Pages for interfacing with users through the browser, and Java servlets for the execution of actions requested. The corresponding model when using Gaelyk is Gaelyk templates and groovlets. The platform is mainly designed for applications employing the request and response usage model. App Engine calls servlets with a request and a response object and waits for the servlet to populate the response object and return. The data in the response object is then sent to the user. It is not possible to perform parts of an execution and send the results to the user, then perform the rest of the execution and send data to the user again. This form of streaming is not supported by App Engine.

Java web applications use a deployment descriptor and consequently so does App Engine. The file is used to map URLs to servlets and JSPs. The deployment descriptor can also be used to restrict access to certain URLs. For example, some page can be made available only to logged in users, App Engine will then automatically redirect users that are not logged in to a login page and then back to the original page after the user logs in. The access restriction can also be specific to administrators of the application. Error handlers can also be defined to control what is sent to the user when errors occur, although some error conditions cannot be customized e. g. the HTTP error codes 403, 404, and 500 (representing quota errors, servlet not found, and internal App Engine error).

App Engine applications use an additional configuration file called `appengine-web.xml`. This file contains information on which files should not be handled as static files since App Engine serves static files such as images and CSS style sheets from dedicated web servers and servlets from other application servers. For a Gaelyk project `.groovy` and `.gtpl` files need to be excluded from the static files. The application id and the version of the application are also specified in this file.

7.3.1 The Users Service

App Engine has a users service that provide support for authentication and authorization using Google accounts. Access to an application can be limited to a custom Google Apps domain or to any one with a Google account. There is also experimental support for OpenID. App Engine also supports the use of different roles; user and administrator. Administrative rights are managed in the web interface of the application. If these features are sufficient for an application the App Engine API can be used to both authenticate users by forcing them to log in and to implement different behavior for administrators and users.

In the case of Wera there may be need for more user roles than just administrator and user. Therefore the supplied support for authentication is used for the application but a custom implementation is used for the authorization. This means that users log on to the application using Google accounts but only users whose Google account has been added as a user in the datastore can actually use the application. The role assigned to the account internally also determines which parts of the application are visible to the user and which actions are allowed.

7.3.2 The Datastore

As described earlier the App Engine datastore stores entities that have different properties. The entities can be arranged hierarchically so that one entity is the parent of a group of other entities. This is important because of the limitations on the datastore. Only entities that belong to the same entity group can be modified or read in a transaction. Entities with the same root ancestor entity belongs to the same entity group. Because of this, entities representing objects of the same kind has been placed in the same entity groups. Entity groups are also a unit of consistency, that is, when querying the datastore queries over many entity groups may return stale results. To guarantee consistency, ancestor queries have to be used.

The properties of the entities can be of all of the common data types like integer, string, and boolean. There is also a text type for longer strings, a blob type for binary data, and more specific App Engine types like the Google Accounts user. Since there is no schema defining the structure of the entities, different entities of the same type can have a different set of properties and different types for properties with the same name. This

gives great flexibility during development. Using an incremental development style features can be added gradually and new entities can be created with new properties without any reconfiguration of the datastore.

Users adding cases to Wera sometimes will want to attach files to them. The datastore has a limitation of 1 MB per entity making it inappropriate for storing files. Instead the Blobstore service is used for this purpose. The Blobstore requires billing to be enabled for the application but there is still a free quota of 1 GB storage once billing is enabled. Using Blobstore the maximum size for files is instead 2 GB which is more than sufficient for attachment files. The files are submitted to the Blobstore through an html input form and can be served back to the users through a web browser.

Chapter 8

Evaluation of Platforms

This chapter presents the results of the evaluation in each of the categories. The points of the platforms and the result for suitability for Wera are presented.

8.1 Cost

The cost of running Wera on the platforms has been calculated for two different migration scenarios. The first scenario is one instance running the database and another running the web server, the second scenario is one larger instance running both. The pricing models for the platforms vary. The most common model for billing computing power is different sizes of instances with hourly rates based on the actual or virtual hardware of the instance. This model is employed in some form by Amazon EC2, Windows Azure, Rackspace Cloud, Flexiant Flexiscale, Atlantic.Net, and City Cloud. OpSource has a finer grained model where the user can choose the exact amount of RAM, storage and number of CPU cores to be used. The pricing model of GoGrid is based solely on the "RAM hours" consumed, other configurations like the number of CPU cores used cannot be adjusted by the user. Google App Engine has no concept of an instance rented by a user but rather charges for the actual CPU utilization in the Google infrastructure. Joyent and Salesforce charges by the month instead of by the hour. Joyent charges for differently sized instances and Salesforce per user of the application.

The prices for network traffic and storage also vary between the vendors. The most common is a fee per GB transferred, many have different fees for outbound and inbound traffic and some charges only for outbound traffic. Some vendors offer free network traffic up to a certain amount and some offers variable prices, where higher amounts of traffic grants a lower fee per GB.

Since the actual usage of the application cannot be measured exactly estimations of likely usage scenarios have been made.

The minimum specifications for the instances of the first scenario are:

- 2 GB RAM
- 2 CPU cores
- 45/40 GB hard drive storage
- Windows Server operating system

For the second scenario, the minimum specifications are:

- 4 GB RAM
- 4 CPU cores
- 45 GB hard drive storage
- Windows Server operating system

The estimations made of the usage for Wera are the following:

- A project with 2000 cases stored, cases have an average of 1 attachment with a size of 0.5 MB, the average size of a case is 1 MB making the size of the database 2 GB.
- 2 cases are added each day and each case is edited 7 times.
- The inbound and outbound network traffic for the application are both 1 GB per month.

Salesforce charges by the number of users of each app, 15 users have been used as an approximation for the calculations. All of the minimum specifications above are not applicable for all platforms and have therefore been ignored in some cases. Tables A.1 and A.2 in appendix A contains all of the prices used in the calculation, figure 8.1 shows the final result.

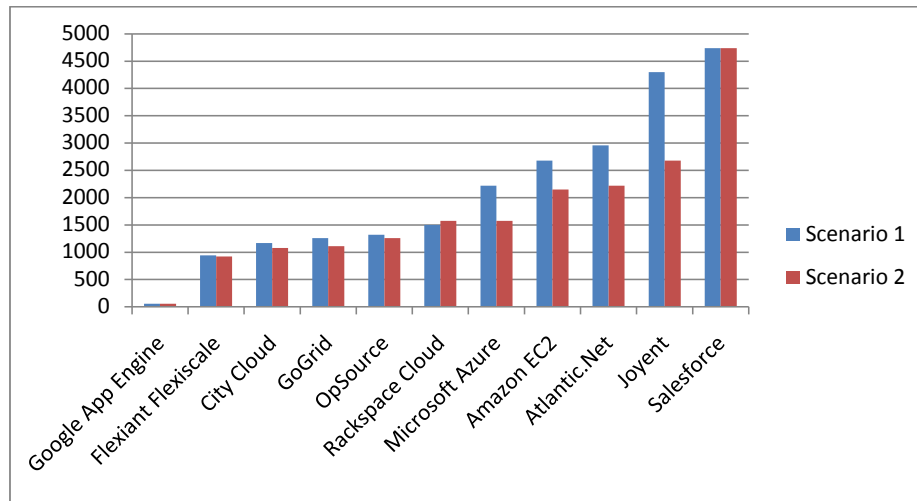


Figure 8.1: The cost of running the two scenarios on the platforms in SEK/month. (Exchange rates from 10/06/2011 when 1£ bought 10.26SEK and 1\$ bought 6.32SEK)

Remarks

The application currently uses Windows SQL Server which requires a license to be used. This license cost has not been taken in regard for several reasons. Firstly, the user may or may not already have a valid license. Secondly, Windows SQL Server cannot be installed on all platforms, applying the cost to just some of the platform will skew the comparison. Finally, the database could just as well be implemented with another database manager. Still, it is important to note that licensing costs have to be taken into regard when making

estimations of the cost of a migration since they can contribute a lot to the final result. The nature of the application is not very business critical and some downtime is not likely to be devastating for the users; therefore price has been prioritized over availability for the comparison.

8.2 SLA

The SLAs of the vendors have some common denominators, but are also very different in some ways. All of the SLAs state that the compensation for any failure cannot exceed the monthly fee for the service, except for City Cloud where the limit is 50 % of the monthly fee. The compensation can also not be received as a payment; it is rather a credit that is used to pay for future use of the service. Most of the vendors state that events of force majeure, DDOS or other attacks, failure of the internet and such is not covered by the SLA. Almost all vendors also exclude scheduled downtime from the uptime guarantee of the SLA. Many of them will not grant credits if the customer has any unpaid bills.

There are different models for compensating for downtime. Some vendors apply a percentage compensation, for example 5 % on the monthly bill for each half hour of downtime. This is the most common model. However, some vendors give themselves some amount of time to fix problems. This means that outages shorter than a specific limit do not grant compensation. Downtime can also be measured differently; some vendors define it as the percentage of five minute periods that the service has been unavailable. These two principles protect vendors from having to grant compensations for many short outages.

There are also added requirements that have to be fulfilled in order for some of the SLAs to be effective. Microsoft's SLA only applies if the user has two or more instances running. Amazon EC2's SLA only applies if more than one availability zone in the same region is unavailable. GoGrid's SLA regarding network downtime only applies if the user has a monthly pre-paid plan, pay-as-you-go users are not covered.

The SLAs has been evaluated in three categories: compensation for hardware outages, compensation for network outages, and whether the SLA includes scheduled downtime. The compensation for downtime is considered more important and therefore has been weighed the highest. The compensations have been compared with respect to downtimes of ten minutes, one hour and ten hours.

Salesforce has no official SLA but a customer can negotiate to get one which may or may not succeed. This may be a disadvantage for smaller companies that do not have the same bargaining position as larger corporations. Google App Engine does not offer any SLA, however Google App Engine for Business will.

Scores have been calculated by awarding points for the compensation offered for hardware and network downtime and the amount of scheduled downtime the provider specifies. The providers have been ordered in each of these categories and been dealt points according to their placings. Table 8.1 shows the sums from the categories, therefore the scores have no meaning by themselves but indicates and the quality of the SLAs when compared to each other. The comparisons of the SLAs can be reviewed in appendix B.

Platform	Score
Rackspace Cloud	39
Joyent	34.5
Flexiant Flexiscale	32
Microsoft Azure	24.5
Atlantic.Net	23.5
OpSource	22.5
GoGrid	18.5
City Cloud	16
Amazon EC2	14.5
Salesforce Force.com	n/a
Google App Engine	n/a

Table 8.1: The score awarded for the SLAs.

Remarks

The SLA is important when evaluating platforms but the limit of compensation at a monthly bill makes the compensation likely to be small compared to the actual loss for a customer in the case of downtime. This means that the actual uptime of a platform is more important than the guaranteed uptime and compensation offered in the SLA. However, this can only be measured by monitoring the platforms over a longer period of time and would come with a substantial cost.

8.3 Security

The subject of security in computing can mean many things and the security regarding cloud computing is not more easily defined due to the ambiguousness surrounding the term *cloud*. Still, security is arguably the most common reason for not taking the step into the cloud. In [16] the authors investigate which of the common security concerns that are actually related to cloud computing and which are not. They argue that problems like downtime, phishing, data loss, and compromised hosts running botnets that have been described as "cloud security" are not directly related to cloud computing since the same problems exists in traditional web application- and data-hosting. The most important security problems arising in cloud computing that are actually new is instead the complexities of multi-party trust considerations and the need for mutual auditability.

In [14] the most important classes of cloud-specific risks identified are:

- Loss of governance
- Lock-in
- Isolation failure
- Compliance risks
- Management interface compromise
- Data protection
- Insecure or incomplete data deletion

- Malicious insider

The largest providers are generally further ahead in addressing these risks but all of the cloud providers still have work to do to in providing the users information regarding these issues. Therefore, evaluating the security of the platforms by reviewing the information made available by the providers was not a successful method. If a larger scale system is to be moved to a cloud platform, each of these potential risks has to be taken into account. Each particular system may be sensitive to different sets of the listed risks, and extensive research may be required to make sure all security policy and legal issues can be resolved.

Below follows a compilation of the information regarding security that the vendors do provide for the platforms and underlying hardware and data centers.

Atlantic.Net does not provide any information on the virtualization platform used. The datacenter features multiple redundant Internet connection, UPS system as well as biometric security for physical access assuring the availability of the service. The data is stored on redundant infrastructure and backed up each night.

City Cloud does not provide much information on the security offered by their platform, they instead refer to Enomaly ¹, provider of the virtualization platform used. The availability of the service is assured through three separate connections from different internet providers and UPS systems to keep the servers up during power outages. The data center is also securely located in a remodeled vault preventing unauthorized physical access.

The Flexiant platform adds network security through the use of VLANs for customers. Customers can also configure a firewall to protect their network through the control panel or the API. Data can be backed up with snapshots and the data center meets tier III standards and has multiple connections to the Internet through different service providers.

GoGrid operates data centers featuring advanced monitoring systems and redundant power supplies and Internet connections. The separate data centers can also be used to provide disaster recovery and fail over capabilities. The firewall available as an add-on feature allows management of VLAN and VPN accesses to Cloud Servers. The GoGrid exchange makes third party images with for example intrusion detection systems available.

Google have adapted the Java runtime and Python interpreter to run in a sandbox environment. Many native C Python modules and Java Native Interface are disabled with Google App Engine. Java applications can also not spawn threads or write data to the local file system. All this is done to be able to keep the different applications isolated and not interfere with the performance and scalability of other applications. Other details about the security concerning Google App Engine are not disclosed to the public. [12]

Joyent provides VLANs and firewalls with hardware load balancers to secure the networks. SmartOS isolates the memory, network, and CPU resources of customers. It also prevents network reconfiguration and traffic sniffing.

OpSource states that data stored in Cloud Files is encrypted with 256-bit AES encryption at rest and 128-bit SSL encryption in-flight. The Cloud Networks offered provides users with VLANs and ACL-based firewall rules. Cloud Networks also offers VPN access and intrusion detection systems. The data centers hosting the OpSource cloud meet tier III standards for data centers set by the Uptime Institute².

Rackspace does not supply much information on the security of their cloud platform other than how data is stored redundantly. To prevent data loss Cloud Files replicate data to different devices and Cloud Servers stores data with RAID-10 and employs snapshot capabilities.

¹<http://www.enomaly.com/>

²http://professionalservices.uptimeinstitute.com/UIPS_PDF/TierStandard.pdf

Salesforce provides extensive information regarding the security of their services. The data centers have redundant power supplies and internal networks among other security measures. The network is protected by both internal and perimeter firewalls and intrusion detection systems. Data is backed up according to a schedule and disaster recovery can be performed since data is spread between different data centers. Salesforce also provides training and guidelines for writing secure applications for Force.com

8.4 Support and Documentation

The level of support offered by the vendors varies. Four of the vendors offer 24/7 support by telephone and live chat or mail while some have no such support. The support is included in some of the offerings while it is a premium feature in some. There are also different support levels available for different prices or based on the service bought by the customer for some of the platforms. The rating for the vendors has been based on the availability of the support and diversity in support channels. Table C.1 shows an overview of the support offered for the platforms.

The documentation of the platforms has been reviewed to find differences and similarities. Nearly all of the vendors offer FAQs with short but informative answers to many common questions. Many vendors also offer forums and wiki styled information sources. Some vendors offer support through the forums where support technicians help customers with support cases. FAQs, forums and wikis have been rated based on comprehensiveness, activeness and clearness. Points have also been awarded for other resources that may be of value to customers. The points awarded to the platforms for the different documentation categories can be found in table C.2, table 8.2 shows the combined result for the platforms.

Platform	Points support	Points documentation	Points total
GoGrid	9.5	7	16.5
OpSource	8	7	15
City Cloud	11	2.5	13.5
Microsoft Azure	3	10.5	13.5
Amazon EC2	2	10.5	12.5
Rackspace Cloud	9.5	2.5	11.5
Salesforce Force.com	4	7	11
Flexiant Flexiscale	5.5	4.5	10
Google App Engine	1	9	10
Joyent	5.5	4.5	10
Atlantic.Net	8	1	9

Table 8.2: The points awarded to the platforms for support and documentation.

Remarks

It is natural that platforms with more users have larger developer communities and more active forums. It is also true that platform-as-a-service offerings are more extensive in functionality. This creates a greater need for documentation and user guides. These factors may skew the comparison to benefit larger platforms and PaaS offerings. There is a trend that larger vendors offer greater resources of articles, whitepapers, and user guides but less support than smaller ones. This makes sense since developing these resources come with a

high onetime cost but can then serve any number of customers in need of information about the platform. The cost of customer support through mail, phone, and chat is on the other hand proportional to the number of customers.

8.5 Complexity of Migration

The complexity of a migration can only really be evaluated when actually performing a migration. Some differences for migrations to platforms are apparent though. Applications can be run on IaaS platforms that support the techniques used without much altering of the application. This is generally not the case with PaaS platforms. When migrating an existing application, PaaS offerings will require more work in adapting the application or even require that the application is re-implemented completely.

However, migrating an application to an IaaS platform may also require some work. Really taking advantage of the capabilities of the cloud platforms like elastic horizontal scaling requires a lot of the design of applications. The load on the system has to be distributed over different nodes that work in parallel, and access to storage must be synchronized in some way. Some platforms have storage services, and to use them for databases or other storage purposes the application has to be integrated with the services. These specialized storage services often provides some backup functionality. If no storage service is available or used, some other backup solution has to be worked out. Other possible features like automatic scaling will also require tuning and adaption of the application. However, the existence of such features should of course be seen as an advantage and not solely a source of extra work.

The conclusion drawn from migrating Wera to IaaS platforms is that the complexity does not differ like it was expected to. In fact, the complexity of the migration processes are so much alike that ranking them by complexity would boil down to making distinctions between small details. This would likely not produce a fair result for the evaluation. The complexity of performing a migration to Azure (and App Engine) is higher though. To reflect this in the evaluation all IaaS platforms has been granted the same points and the PaaS alternatives slightly lower points.

8.6 Results

The evaluation has resulted in points for each platform in each category except for security. Table 8.3 shows these points.

With this information in hand the question is how to employ it. For Wera, my supervisor has acted as a buyer and assigned weights to the categories corresponding to the importance for the application. The weights are shown in figure 8.2 and the result for each platform with the weights assigned to the points can be viewed in table 8.4. In this way, weights can be assigned for any application or system and the points used to generate suggestions of suitable platforms. Specific requirements may disqualify some platforms or make others more attractive but the model may at least give a general idea for which platform to use. It is also important to notice that although security was deemed to be the least important criteria for Wera, there is still a bare minimum that has to be fulfilled by all platforms to be considered at all.

Platform	Cost	SLA	Documentation&Support	Complexity
Amazon EC2	3	3	7	7.5
Atlantic.Net	2	7	1	7.5
City Cloud	9	4	8.5	7.5
Flexiant Flexiscale	10	9	3	7.5
GoGrid	6.5	5	11	7.5
Google App Engine	11	1	3	1.5
Joyent	4	10	3	7.5
Microsoft Azure	8	8	8.5	3
OpSource	6.5	6	10	7.5
Rackspace	5	11	6	7.5
Salesforce Force.com	1	2	5	1.5

Table 8.3: The points awarded to the platforms in each of the categories.

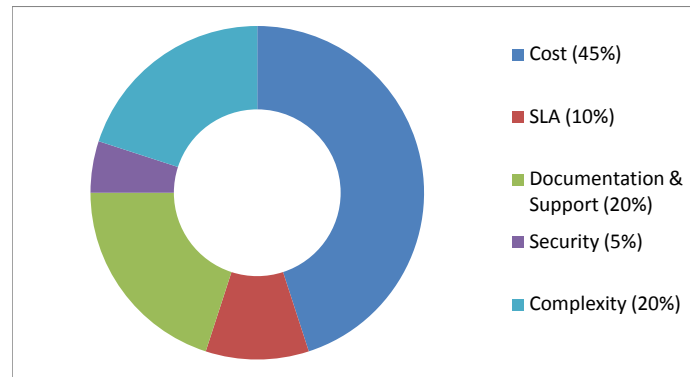


Figure 8.2: The weights assigned to the categories for Wera.

Platform	Result
City Cloud	7,65
Flexiant Flexiscale	7,5
GoGrid	7,125
OpSource	7,025
Microsoft Azure	6,7
Rackspace	6,05
Google App Engine	5,95
Joyent	4,9
Amazon EC2	4,55
Atlantic.Net	3,3
Salesforce Force.com	1,95

Table 8.4: The points awarded to the platforms calculated with the assigned weights for Wera.

Chapter 9

Conclusions

The goals set for the thesis has been reached, an evaluation of cloud computing platforms has been performed and Wera has been migrated to most of them to test the platforms and the migration processes in practice.

The result of the evaluation is that the Swedish provider City Cloud is the most suitable alternative for Wera. The points awarded to the platforms can also be used to produce suggestions for other application by assigning weights to the different categories according to the needs of the application. As for a recommendation for whether to migrate Wera or not, the conclusion is that a migration would not be beneficial, at least for the moment. A cloud platform is definitively a good hosting alternative for the application but the most important aspect for the hosting of Wera is cost. Since the hosting needs is today provided by existing infrastructure a migration would not yield any significant savings making the cloud alternative more expensive. However, if the need arises to upgrade the current hosting, a cloud platform may very well be a good alternative. Since Wera is an application with relatively low traffic, the granularity of billing on an IaaS platform may be a disadvantage, at least one instance would always need to be running and the utilization may be low. A finer grained billing model, like that of Google App Engine, may be better suited. Though, in the case of App Engine the application would need to be redeveloped which is likely to cancel out potential savings.

An experience gained from performing migrations to several IaaS platforms is that they are very much alike. The storage models and features available may differ but the functionality offered is essentially the same. The pricing models and structure of the service level agreements are also very similar. Additionally, the fact that the area is still new is very visible when working with the platforms. Many providers have added features or changed their offering in some way during the course of this project and many features are still in beta stage. However, even though the platforms are still evolving, they are useful. Disruptions in the availability are rare and it is surprisingly easy to migrate an application to an IaaS platform and have it run in the cloud. Extracting the full value of platforms integrating with the storage solutions available on some platforms and employing the scalability requires more work and puts more demands on the applications, but may well be worth the effort. Employing the PaaS offerings also requires a greater effort to get started but when the system is in place there is even more to gain by tasks like patching and automatic scaling being transferred to the provider.

9.1 Limitations

A limitation of this project is the lack of a result in the evaluation of security of platforms. Reaching a viable result would require a more advanced method of evaluation, possibly putting the security mechanisms to the test in practice, which could be a whole project in itself.

9.2 Future Work

Cloud computing is by many predicted to grow and become more widely employed and there is a lot that can be done to evaluate the services further. The performance aspect is important and could be studied in depth. A study would need to consider all aspects of performance on cloud platforms including the performance of I/O, CPU on instances, latency, and transfer rate of the network connection. To get reliable results, the performance also needs to be measured over time. The availability over time is also something to be investigated.

Cloud middleware, or cloud management platforms, is another aspect of cloud computing, raising the level of abstraction and enabling users to leverage multiple cloud platforms. Evaluating alternatives in this area is another direction of future work.

Chapter 10

Acknowledgements

I would like to thank my supervisor at Sogeti Jonas Eklund for his help and appreciated input throughout the work on the thesis. I would also like to thank the people at Sogeti for making my time at the office enjoyable. In addition I would like to thank my supervisor at Umeå University Mikael Rännar for help with this report. Finally, I would like to thank my wife Josefin for proofreading the report and for her support during the project.

References

- [1] Company overview. <http://www.atlantic.net/About-Us/company-overview.html> (visited 2011-04-20).
- [2] What is aws? <http://aws.amazon.com/what-is-aws/> (visited 2011-04-20).
- [3] Amazon elastic block store (ebs). <http://aws.amazon.com/ebs/> (visited 2011-04-20).
- [4] Amazon elastic compute cloud (amazon ec2). <http://aws.amazon.com/ec2/> (visited 2011-04-20).
- [5] Amazon elastic compute cloud faq (amazon ec2). <http://aws.amazon.com/ec2/faqs/> (visited 2011-04-20).
- [6] Amazon relational database service (amazon rds). <http://aws.amazon.com/rds/> (visited 2011-04-20).
- [7] Amazon simple storage service (amazon s3). <http://aws.amazon.com/s3/> (visited 2011-04-20).
- [8] Amazon simpledb. <http://aws.amazon.com/simpledb/> (visited 2011-04-20).
- [9] Amazon simple queue service (amazon sqs). <http://aws.amazon.com/sqs/> (visited 2011-04-20).
- [10] Amazon virtual private cloud (amazon vpc). <http://aws.amazon.com/vpc/> (visited 2011-04-20).
- [11] L. Badger, T. Grance, R. Patt-Corner, and J. Voas. Cloud computing synopsis and recommendations. Technical report, National Institute of Standards and Technology, 2011.
- [12] C. Balding. Cloudsecurity.org interviews guido van rossum: Google app engine, python and security, 2008.
- [13] J. Brodtkin. Rackspace challenges amazon with new cloud server, storage services, 2009. <http://www.networkworld.com/news/2009/031309-rackspace-cloud-server-storage.html> (visited 2011-03-06).
- [14] D. Catteddu and G. Hogben. Cloud computing - benefits, risks and recommendations for information security. Technical report, European Network and Information Security Agency, 2009.
- [15] D. Chappell. Introducing the windows azure platform. Technical report, DavidChappell & Associates, 2010.

- [16] Y. Chen, V. Paxson, and R. H. Katz. What's new about cloud computing security. Technical report, Dept. of Electrical Engineering and Computer Sciences, University of California at Berkeley, 2010.
- [17] City network lanserar den första skandinaviska cloud computing tjänsten för den europeiska marknaden - www.citycloud.eu, 2010. http://www.mynewsdesk.com/se/pressroom/webbhotell-city_network/pressrelease/view/city-network-lanserar-den-foersta-skandinaviska-cloud-computing-tjaensten-foer-den-europeiska-marknaden-www-citycloud-eu-395314 (visited 2011-03-06).
- [18] M. Armburst et al. A view of cloud computing. *Communications of the ACM*, 53(4):50–58, 2010.
- [19] About flexiant. <http://www.flexiant.com/about/> (visited 2011-03-06).
- [20] I. Fried. Microsoft launches windows azure, 2008. <http://news.cnet.com/microsoft-launches-windows-azure/> (visited 2011-03-06).
- [21] Gaelyk - a lightweight groovy toolkit for google app engine java. <http://gaelyk.appspot.com/> (visited 2011-06-09).
- [22] Gogrid continues global expansion with new european data center, 2011. <http://www.gogrid.com/about/press-releases/gogrid-continues-global-expansion-with-new-european-data-center> (visited 2011-03-06).
- [23] Företagsöversikt. <http://www.google.com/intl/sv/corporate/> (visited 2011-03-06).
- [24] Quotas. <http://code.google.com/appengine/docs/quotas.html> (visited 2011-03-06).
- [25] Groovy - an agile dynamic language for the java platform. <http://groovy.codehaus.org/> (visited 2011-06-09).
- [26] A. R. Hickey. Amazon cloud outage aftermath: Questions, concerns linger, 2011. <http://www.crn.com/news/cloud/229500034/amazon-cloud-outage-aftermath-questions-concerns-linger.htm?itc=refresh> (visited 2011-06-21).
- [27] D. Hilly. Cloud computing: A taxonomy of platform and infrastructure-level offerings. Technical report, College of Computing, Georgia Institute of Technology, 2009.
- [28] Java servlet technology. <http://www.oracle.com/technetwork/java/javaee/servlet/index.html> (visited 2011-06-10).
- [29] Joyent is a global cloud computing software and services company offering cloud computing solutions worldwide since 2004. <http://www.joyent.com/about/> (visited 2011-03-06).
- [30] Innovative application virtualization. aggressive content caching in memory. just-in-time resource provisioning. joyent virtual machines were developed for the web., 2010. <http://www.joyent.com/documents/Joyent-SmartMachine-and-VirtualMachine-Data-Sheet.pdf> (visited 2011-03-06).
- [31] A. Khajeh-Hosseini, I. Sommerville, and I. Sriram. Research challenges for enterprise cloud computing, 2010.

-
- [32] T. Chamberlin L. Leong. Magic Quadrant for Cloud Infrastructure as a Service and Web Hosting. Technical report, Gartner, 2009.
- [33] L. Lamport. Paxos made simple. *ACM SIGACT News (Distributed Computing Column)*, 32:51–58, 2010.
- [34] L. Leong. How to select a cloud computing infrastructure provider. Technical report, Gartner, 2009.
- [35] P. Mell and T. Grance. The nist definition of cloud computing. Technical report, National Institute of Standards and Technology, 2011.
- [36] B.P. Rimal, E. Choi, and I. Lumb. A taxonomy and survey of cloud computing systems. *2009 Fifth International Joint Conference on INC, IMS, and IDC*, pages 44–51, 2009.
- [37] L. Siegele. Let it rise: A special report on corporate it. *The Economist*, 2008.
- [38] J. W. Simth. A comparison of public cloud platforms - microsoft azure and google app engine. Technical report, St Andrews Cloud Computing Laboratory, 2009.
- [39] T. Singh. Google app engine vs windows azure, 2009. <http://geeknizer.com/google-app-engine-vs-windows-azure/> (visited 2011-06-16).
- [40] E. van Ommeren and M. van den Berg. *Seize the Cloud - A Manager's Guide to Success with Cloud Computing*. IBM and Sogeti, 2011.
- [41] L. M. Vaquero, L. Rodero-Merino, J. Caceres, and M. Lindner. A break in the clouds: Towards a cloud definition. *ACM SIGCOMM Computer Communication Review*, 39(1):50–55, 2009.
- [42] Aws case study: Washington post. <http://aws.amazon.com/solutions/case-studies/washington-post/> (visited 2011-06-08).
- [43] Q. Zhang, L. Cheng, and R. Boutaba. Cloud computing: state-of-the-art and research challenges. *Journal of Internet Services and Applications*, 1(1):7–18, 2010.

Appendix A

Cost Evaluation

Table A.1 lists the cost of the instance types needed for Wera’s migration scenarios. Table A.2 shows the costs of network transfer and storage on the platforms. Salesforce does not have this kind of billing, they charge \$50 per user and month. For Google App Engine there is an always on feature that is deemed necessary for Wera, it costs \$0.3/day.

Flexiant use the concept of ”units”, an account on the platform has a unit balance and all services are charged in units. The price of a unit varies depending on the number of units purchased in a transaction, the smallest amount that can be purchased is 1000 costing £11 and the largest amount is 2000000 costing £18400.

Platform	Instance 1	Instance 2
Amazon EC2	0.29/h	0.48/h
Atlantic.Net	0.32/h	0.58/h
City Cloud	0.8 SEK/h	1.34 SEK/h
Flexiant Flexiscale	9 units/h	15 units/h
GoGrid	99.5/month	199/month
Google App Engine	6.5 CPU hours/day free, then 0.1/h	
Joyent	340/month	340/month
Microsoft Azure	0.24/h	0.24/h
OpSource	0.13/h	0.26/h
Rackspace Cloud	£0.10/h	£0.21/h
Salesforce Force.com	na	na

Table A.1: The cost for the services, the currency is USD unless otherwise stated. Instance 1 corresponds to at least 2 GB ram and 2 CPUs and Instance 2 at least 4 GB ram and 4 CPUs.

Platform	Outbound/inbound transfer	Instance storage
Amazon EC2	1 GB free, then 0.15/GB / 0.1/GB	350/850 ephemeral storage
Atlantic.Net	0.14/GB / 0.05/GB	320/500 GB included
City Cloud	500 GB free, then 0.5 SEK/GB	20 GB free, then 100 SEK/GB
Flexiant Flexiscale	5 units/GB	5 units/GB, 2 units/GB I/O
GoGrid	0.29/GB / free	100/200 GB included
Google App Engine	1/1GB/day free, then 0.12/GB / 0.1/GB	1GB free, then 0.15/GB
Joyent	included	50GB included
Microsoft Azure	0.15/GB / 0.10/GB	490 GB included
OpSource	0.15/GB / free	0.219/GB
Rackspace Cloud	£0.12/GB / £0.05/GB	80/160GB included
Salesforce Force.com	na	na

Table A.2: The cost for the services, the currency is USD unless otherwise stated.

Appendix B

SLA Evaluation

Table B.1 shows a summarization of the SLAs of the vendors. The compensations offered by Microsoft for hardware downtime are 10 % of the monthly bill for downtime exceeding 0.1 % each month and 25 % of the monthly bill for downtime exceeding 1 %. For network downtime, the thresholds are 0.05 % and 1 % of a month for 10 % and 25 % compensation respectively. OpSource offers a compensation of 5 % of the monthly bill per hour for the first three hours of downtime and 10 % for any additional hours of downtime each month.

Platform	Hardware	Compensation	Network	Compensation	Min. dur.
Amazon EC2	99,95 %	10 %	99,95 %	10 %	5 min
Atlantic.net	100 %	5 % per 30 min	100 %	5 % per 30 min	60 min
City Cloud	100 %	5 % per 3h	100 %	5 % per 3h	
Flexiant Flexiscale	100 %	5 % per 30 min	100 %	5 % per 30 min	
GoGrid	100 %	100 x cost	100 %	100 x cost	15 min
Joyent	100 %	5 % per 30 min	100 %	5 % per 30 min	
Microsoft Azure	99,9 %	10 %/25 %	99,95 %	10 %/25 %	5 min
OpSource	100 %	5 %/10 % per h	100 %	5 %/10 % per h	
Rackspace Cloud	100 %	5 % per 30 min	100 %	5 % per 30 min	

Table B.1: The SLA guarantees offered by the vendors, hardware and network percentages are the promised uptime for each month (Amazon measures uptime by the year), and the compensation percentages are based on the monthly cost. Min. dur. is the minimum duration for a failure that grants any compensation.

Hardware	Platform	10 min	1 h	10 h
	Amazon EC2	0 %	0 %	10 %
	Atlantic.Net	0 %	0 %	90 %
	City Cloud	5 %	5 %	20 %
	Flexiant Flexiscale	0 %	10 %	95 %
	GoGrid	0 %	14 %	100 %
	Joyent	5 %	10 %	100 %
	Microsoft Azure	0 %	10 %	25 %
	OpSource	0 %	5 %	90 %
	Rackspace Cloud	5 %	10 %	100 %
Network	Platform	10 min	1 h	10 h
	Amazon EC2	0 %	0 %	10 %
	Atlantic.Net	5 %	10 %	100 %
	City Cloud	5 %	5 %	20 %
	Flexiant Flexiscale	0 %	10 %	100 %
	GoGrid	2 %	14 %	100 %
	Joyent	5 %	10 %	100 %
	Microsoft Azure	0 %	10 %	25 %
	OpSource	0 %	5 %	90 %
	Rackspace Cloud	5 %	10 %	100 %

Table B.2: The compensations offered by vendors for hardware and network failures of 10 minutes, 1 hour, and 10 hours. The percentages are the amount of the monthly bill refunded.

Appendix C

Support and Documentation Evaluation

Platform	Support channels	Availability
Amazon EC2	none included ^a	
Atlantic.Net	phone, mail	24/7
City Cloud	phone ^a	Mon-Fri 08.00-22.00, Sat-Sun 10.00-14.00
Flexiant Flexiscale	phone, support ticket	24/7 critical errors, Mon-Thu 9:00-5:30, Fri 9.00-4:30 GMT
GoGrid	phone, chat	24/7
Google App Engine	none included	
Joyent	phone, chat, mail	24/7 production outage, 8/5 tech support
Microsoft Azure	online form	24/7
OpSource	phone, forum	24/7, forum - reply within 30/120 min
Rackspace Cloud	phone, chat	24/7
Salesforce Force.com	phone ^a	12/5

Table C.1: The support offered for the platforms.

^aPremium support available with 24/7 support.

Platform	FAQ	Forum	Wiki	Other	Score
Amazon EC2	3	3	3	articles, libraries, code samples	3 12
Atlantic.Net	1	na	na		0 1
City Cloud	1	na	na	a few simple guides	1 2
Flexiant Flexiscale	3	na	na	code snippets	0 3
GoGrid	2	1	3		0 6
Google App Engine	3	3	3	articles, cookbook	2 11
Joyent	1	2	na		0 3
Microsoft Azure	3	3	3	whitepapers, code samples gallery	3 12
OpSource	3	3	na		0 6
Rackspace Cloud	2	na	na		0 2
Salesforce Force.com	1	2	2	cookbook, code share	1 6

Table C.2: A rating of FAQ, forum, wiki and other information resources for the platforms. Note that these ratings are subjective and based on my personal opinions.

Appendix D

List of Abbreviations and Acronyms

AES	Advanced Encryption Standard
AMI	Amazon Machine Images
API	Application Programming Interface
AWS	Amazon Web Services
CDN	Content Delivery Network
DDoS	Distributed Denial of Service
EBS	Elastic Block Storage
EC2	Elastic Compute Cloud
FTP	File Transfer Protocol
IaaS	Infrastructure-as-a-Service
IIS	Internet Information Services
JSP	JavaServer Pages
PaaS	Platform-as-a-Service
RDS	Relational Database Service
REST	Representational State Transfer
S3	Simple Storage Service
SaaS	Software-as-a-Service
SCP	Secure Copy
SDK	Software Development Kit
SLA	Service Level Agreement
SOAP	Simple Object Access Protocol
SQS	Simple Queue Service
SSL	Secure Sockets Layer
UPS	Uninterruptible power supply
VLAN	Virtual Local Area Network
VPC	Virtual Private Cloud
VPN	Virtual Private Network
