Technical University of Denmark

02228 Fault-Tolerant Systems

Fault-Tolerant Cloud Computing Architectures

Authors:

Andreas Hallberg Kjeldsen
s092638@student.dtu.dk

Morten Chabert Eskesen
s133304@student.dtu.dk

December 9, 2014

Chapter 1

Introduction

In this report we will describe what cloud computing is, further we will give a detailed description of the architecture and fault-tolerant features of two cloud system, at last we will compare how the systems handle failures and discuss the pros and cons of these methods. As a result of the comparison, we will be able to conclude on what the systems do well and where they might be able to improve.

1.1 Scope

We will focus on the fault-tolerant features of the cloud computing architecture within the two selected cloud computing systems. We have chosen to focus on Amazon Web Services and Google Cloud Platform. We have chosen these cloud computing systems because both systems are among the most popular[1] cloud computing systems [10].

1.2 Cloud Computing

The National Institute of Standards and Technology is a federal technology agency in the United States of America. They define cloud computing by the following:

"Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management effort or service provider interaction." [11]

This definition states that shared networks, servers, applications, services etc can easily be distributed globally and quickly maintained by using cloud computing.

There are five essential characteristics of the cloud computing model

[1] Popular means that they are among the most commonly used platforms for enterprise cloud developers.

On-demand self-service No required human interaction when needing more or less computing capabilities

Broad network access Capabilities are accessed through standard mechanisms and available over the network

Resource pooling Computing resources are pooled in order to serve multiple consumers

Rapid elasticity Capabilities can be elastically released to scale rapidly according to the demand

Measured service In an automatic way the cloud computing systems control and optimize resource use

There are four different deployment models. One cloud infrastructure is for exclusive use by a single organization comprising the multiple consumers - the *private cloud*. Another cloud infrastructure is for exclusive use by a specific community of users from organizations with shared concerns - the *community cloud*. The *public cloud* is an infrastructure open for use by the general public. The last cloud infrastructure is a mixture of two or more distinct cloud infrastructures that remain unique entities - the *hybrid cloud*.

1.3 Fault tolerance in cloud computing

Fault tolerance is a key factor for cloud computing systems due to the rapid exponential growth in use of cloud computing [12]. The purpose of fault tolerance in any system is to achieve robustness and dependability. Fault tolerance policies and techniques allow us to classify this techniques into 2 types

Proactive fault tolerance policy aims to avoid recovering from fault, errors and failure by predicting them and replacing the suspicious component. This means detecting problems before they actually occur.

Reactive fault tolerance policy reduces the effect of failures when the failure actually occurs.

These policies can be divided into two further sub techniques error processing and fault treatment. The aim of error processing is to remove errors from the computational state and the aim of fault treatment is to prevent faults from reoccurring.

Chapter 2

Amazon Web Services

The Amazon Web Services, henceforth *AWS*, is a collection of remote computing services which make up a cloud computing platform. AWS was launched by Amazon.com in 2006. This chapter will outline the architecture of their platform and describe the fault-tolerant features the platform provide.

2.1 Architecture

AWS offers a wide range of services, which can be seen as components of a system. The services they provide are available in multiple continents. The services can be used to build entire systems which are highly fault-tolerant. Using AWS the hassle of obtaining servers in various locations is removed, instead the user just has to select which region to deploy to. Though it might be tempting to just push an entire system out into the cloud, it is not enough to make it fault tolerant. The architecture of the system is important.

2.1.1 Basic architecture

In Figure 2.1 is a system with two availability zones, each hosting the web server, application server and a database server. Replication is setup between the databases. Dependencies between the availability zones has been avoided. The load balancing makes sure to distribute the traffic properly. If one of the web servers or application servers fail, then the load balancer could redirect all traffic to availability zone without errors. Though it's a way to handle the fault, it is not always desirable to reroute all traffic as this can cause the servers to be overloaded with traffic. The system is also prone to downtime in case of multiple failures, such as an availability zone being unavailable while a component in the other availability zone crashes. In this case, there could be a data loss and the system wouldn't be available.

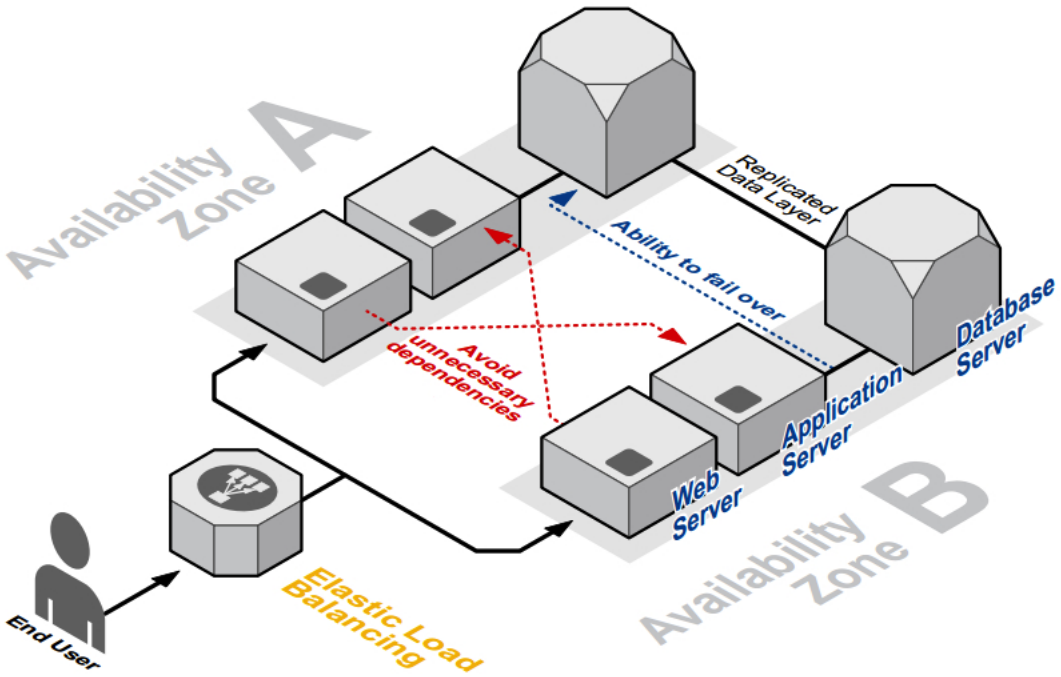

Figure 2.1: Example of a system which has some fault-tolerance but is prone to downtime in case of multiple failures [1].

2.1.2 AWS architecture

Using the services from AWS to build a system, makes the components of the system have some fault-tolerance, but that is not always enough. Luckily, AWS is made to make fault-tolerance easy. In Figure 2.2 is a system with two availability zones, each hosting an application instance and a storage instance. At specified time intervals, snapshots of the storage instances are saved to another storage system, which also has fault-tolerance implemented.

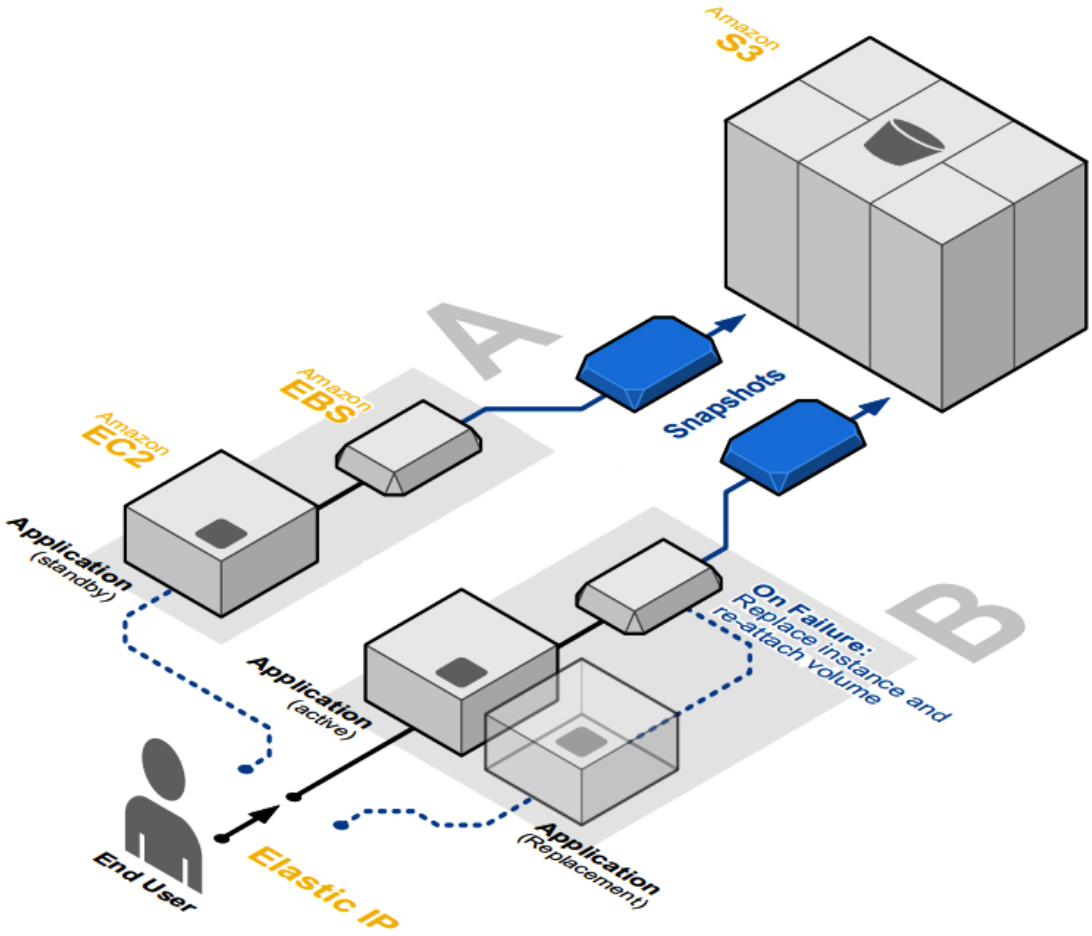

Figure 2.2: Example of a system hosted on AWS with fault-tolerance [1].

2.2 Geographic Availability

AWS have regions in various parts of the world. Each region has multiple Availability Zones, henceforth *AZ*. When using AWS, the user has to specify in which region they wish to have their resource[1] located. Within the region, a resource may then be distributed to multiple AZ.

[1] A resource can be a computing resource, storage resource etc.

2.3 Redundant storage

AWS provides multiple ways of storing data. They have key/value based storage (S3), block storage (EBS) and physical SSDs.

2.3.1 Simple Storage Service

The Amazon Simple Storage Service, henceforth *S3*, is a highly-scalable, reliable and low-latency key/value data storage infrastructure. With S3 you define buckets in which you wish to store data. Each bucket range from 1 TB to 5TB in capacity and has a maximum file size of 5 GB. Within an AZ, the data stored in S3 is replicated to various data centers. Storing the data within various data centers, allows S3 to fetch the data requested from whichever data center that contains the requested data with the lowest latency.

2.3.2 Elastic IP

2.3.3 Load Balancing

2.4 Fault-Tolerant Features

Chapter 3

Google Cloud Platform

3.1 Architecture

3.2 Fault-Tolerant Features

Chapter 4

Comparison of Failure Handling

List of faults that the systems handle along with a description of how it's handled and why it works. If the methods for handling the failure differ, we will discuss the methods, highlighting their pros and cons.

Chapter 5

Conclusion

Conclude on our findings, focus on what the systems do well and where it might be possible to improve.

Bibliography

[1] Amazon Web Services Reference Architectures, *Fault Tolerance & High Availability*, 2014. http://media.amazonwebservices.com/architecturecenter/AWS_ac_ra_ftha_04.pdf

[2] Amazon Web Services Whitepapers, *Building Fault Tolerant Applications*, October 2011. http://media.amazonwebservices.com/AWS_Building_Fault_Tolerant_Applications.pdf

[3] Amazon Web Services, *Designing Fault-Tolerant Applications*, Slides, July 2011. http://www.slideshare.net/AmazonWebServices/base-camp-awsdesigningfaulttolerantapplications

[4] Amazon Web Services, *Designing Fault-Tolerant Applications*, YouTube, July 2011. https://www.youtube.com/watch?v=9BrmHoyFJUY

[5] Google, *Google Cloud Platform Documentation*. https://cloud.google.com/docs/

[6] Google Patents, *Data placement for fault tolerance*, February 2006. http://www.google.com/patents/US7000141

[7] Google I/O, *App Engine: Scalability, Fault Tolerance, and Integrating Amazon EC2*, YouTube, June 2006. https://www.youtube.com/watch?v=p4F62q1kJ7I

[8] Google, *Google Cloud Platform Blog*. http://googlecloudplatform.blogspot.dk/

[9] Todd R. Weiss, *Google Cloud Platform Gets Developer Enhancements*, August 2013. http://www.eweek.com/cloud/google-cloud-platform-gets-developer-enhancements

[10] Larry Dignan, *Amazon Web Services, Windows Azure top cloud dev choices, says survey*, August 2013. http://zd.net/1vw5pzE

[11] National Institute of Standards and Technology, *The NIST Definition of Cloud Computing*, October 2011. http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf

[12] Prasenjit Kumar Patra, Harshpreet Singh and Gurpreet Singh, *Fault Tolerance Techniques and Comparative Implementation in Cloud Computing*, February 2013. International Journal of Computer Applications.

[13] Terrell Herzig et al., *Implementing Information Security in Healthcare: Building a Security Program*, HIMSS, February 2013