Vrije Universiteit Amsterdam

Honours Programme, Project Report

Failure Analysis of Big Cloud Service Providers Prior and During Covid-19 Period

---

Author: Muhammad Ahsan (2663138)

1st supervisor:	prof. dr. ir. Alexandru Iosup
daily supervisor:	Sacheendra Talluri
2nd reader:	1st supervisor & daily supervisor

A report submitted in fulfillment of the requirements for the Honours Programme, which is an excellence annotation to the VU Bachelor of Science degree in Computer Science/Artificial Intelligence/Information Sciences
version 1.0

May 11, 2022

Abstract

Cloud services are important for societal function such as healthcare, commerce, entertainment and education. Cloud can provide a variety of features such as increased collaboration and inexpensive computing. Failures are unavoidable in cloud services due to the large size and complexity, resulting in decreased reliability and efficiency. For example, due to bugs, many high-severity failures have been occurring in cloud infrastructures of popular providers, causing outages of several hours and the unrecoverable loss of user data (1, 2). There are limited studies that show cloud service failure analyses using various sources such as news articles. However, a detailed cloud failure focused study is required that provides analyses for cloud failure data gathered directly from the vendors. Furthermore, the Covid-19 cloud failures should be studied as cloud services played a major role throughout the Covid-19 period, as individuals relied on cloud services for activities such as working from home. A program can be made for this task. As a result, we will be able to better understand and mitigate cloud failures to reduce the effect of cloud failures.

Contents

1 Introduction **1**

1.1 Context . . . 1
1.2 Problem Statement . . . 2
1.3 Research Questions . . . 3
1.4 Approach . . . 4
1.5 Main results and contribution . . . 5

2 Background **6**

3 Results and Analysis **8**

3.1 Cleaning Data . . . 8
3.1.1 Duplicates . . . 8
3.1.2 Unknown cells and data limitations . . . 9
3.1.3 Incorrect dates . . . 9
3.1.4 Special case rows . . . 10
3.1.5 Different words representing same meaning . . . 10
3.1.6 Organising Data . . . 12
3.2 Analysis . . . 13
3.2.1 Monthly Plot . . . 13
3.2.2 Comparing Vendors . . . 14
3.2.3 Failure time and recovery . . . 15
3.2.4 Cloud Services Failures . . . 16
3.2.5 Time of Cloud Failures . . . 18
3.2.6 Location of Cloud failures . . . 21

4 Evaluation and Related Work **22**

4.1 Analysis of Limitations . . . 22
4.2 Related Work . . . 22

CONTENTS

5 Conclusion **24**

5.1 Conclusion and Future Work 24

6 Self-Reflection **25**

6.1 Self-Reflection 25

References **26**

1

Introduction

1.1 Context

Cloud Computing is a model for enabling 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. The cloud is ubiquitous. It is clear that utilizing the cloud is a trend that continues to grow as it is used by an increasing number of individuals and businesses (3, 4). Users may utilize cloud services when and where they need them and only pay for the resources they use. It provides massive computational capacity, scalability on demand, and utility-like availability at a reasonable cost. However, cloud failures occur frequently and affect users of the cloud service.

A failure can be described as an unintended behavior of a service. The term "cloud failure" refers to when a cloud service is unavailable for use for an extended period of time. Unavailability can also relate to a service's inadequacy in terms of performance, as measured by the agreed-upon SLA metrics. An SLA is an explicit or implicit contract between a cloud provider and customer. It governs the obligations and responsibilities between both parties regarding the provided service. In case the cloud provider is unable to meet the terms of the agreed SLAs, the consequence is usually a financial penalty or rebate, but it can take other forms as well. For example, if a data centre was only partially impacted by a failure, the vendor may be forced to do the appropriate repair and restoration actions. The end-user may experience downtime until the service is fully restored in accordance with the agreed-upon SLA criteria. Despite all the potential and development cloud computing has undergone over years, failures continue to occur. Cloud failures are of concern as they can result in reduction of quality of Service, availability,

reliability and energy-waste (5) that can ultimately lead to economic loss for both cloud users and providers. By studying the failures we can protect users from being affected by these failures.

There are numerous cloud providers. Cloud failures are common among all providers. Big cloud companies represent a large population of cloud users. Therefore, it is more beneficial to study failures in big cloud providers. This is the aim of the current study. Many big cloud vendors provide cloud failure logs that can be used for the study. Cloud failures can be compared over time to gain better understanding. Furthermore it is interesting to compare failures among different vendors. Cloud failure analysis is done with the aid of a program.

There are studies that describe different aspects of cloud. However there is limited in depth research in the topic of cloud failures. This paper will provide analysis about cloud failures in three popular big cloud providers Amazon Web Service (AWS), Microsoft Azure and Google Cloud Platform (GCP), and comparison of failures during the time period 2018 till 2020 (June).

1.2 Problem Statement

As the dependency on cloud computing increases, society demands high availability, an ideal 24/7 service up-time, if possible. Even under normal world conditions, the cloud runs many daily life activities including banking, healthcare, governance, transportation, e-commerce, entertainment, etc.

Cloud services played an important part during Covid-19. A large population of the world blindly trusts cloud services to run their daily lives, for example, employees working remotely from homes, online education for students and online businesses. However, many cloud services continue to fail. These failures can lead to great loss as people's daily activities, especially many people's income depend on the cloud services to work.

Is it correct to depend on the cloud to run daily lives? Why does the cloud keep failing.? Which services are failing? Where, when and how are the failures occurring? How do the big cloud providers respond to failures? What is the pattern of the failures in different time periods? During difficult times like the Covid-19 period, can the cloud fulfil the needs of people. Many similar important questions can be explained by analysing the failures on cloud.

1.3 Research Questions

[RQ1] How to process raw cloud failure data to carry out analysis using the data?

The amount of data is substantial, since most files include approximately 1000 instances. Any manual approach is likely to have mistakes and is inefficient for solving the current problem because data is raw and would require customize-able processing. Therefore an efficient method to solve the issue of preparing data is to use an automated method which is reproducible. One of the possibilities is to use a program that does the task. However, no known program exists that can prepare the given sets of data.

[RQ2] How to transform processed data for data visualization and extract useful information? In what ways can cloud failure data be visualized?

After approaching research question 1, we would have a software that outputs the final data sets. At this point some approach is required that can extract information and present the data. A possibility can be to provide the processed data as an input to some available data analysis software. Some softwares like Microsoft Excel only provides basic ways to analyze and do not meet the requirement of the current study for example if the study requires a specific graph with specific features, the data analysis software may not have those. Additionally, there are three different data sets, doing manual analysis using some software for every data file is not efficient. In a nutshell, general data analysis software provide limited options.

[RQ3] Based on the information extracted, what analysis can be made? What can we learn by comparing the Covid-19 period with prior years?

Information such as peak cloud failures and their relation with other features such as time and location need to be extracted to gain valuable insight in cloud failures. Additionally, People relied on cloud services during Covid-19 example doing work from home. The change in usage can help us view failures patterns that differ from normal days. Therefor comparison of Covid-19 period cloud failures with period before before can be used to better understand cloud failure.

1.4 Approach

RQ1. The question under consideration can be solved by creating a program that takes the raw data file as the input, prepares it such as cleaning, filtering, organising e.t.c. The program outputs a file with ready to use data. There are a variety of programming languages available for making the software. A preferable language can be python because there are many data-centric Python packages, for example Pandas and NumPy, which make the process of data processing and data analysis a lot quick and convenient.

RQ2. The current research requires an analysis approach that is efficient, reusable and provides a wide range of customizing options. The given requirements can be fulfilled by coding a software that does the job. This software can be an extension of the program of research question 1. In this way the data can directly be available for analyses. Moreover, as suggested in the approach to research question 1, python can be used as it provides a variety of options example for plotting matplotlib and seaborn plot can be used.

RQ3. The first part of the question can be approached by understanding the general structure of the data files such that considering the data fields(columns of data set), it is possible to predict some possible analyses. The yearly data can be grouped by separate vendors. This can provide answers to questions like which vendor has the most failures. The event start time and date of failures can be used to tell which month had the most failures. Using event start time with event end time, the following can be answered which event lasted the longest and how long was the maximum duration. This can further be used to analyse the performance of individual service providers. The location data can tell which location had the most failures with respect to individual vendors. Similarly service name column data can tell which services failed the most with respect to individual vendors. A possible plot can be a plot showing months vs failure count for each month with respect to separate cloud providers.

The second part of the question can be answered such that, the Covid-19 reduced the physical interaction among people and increased interaction through cloud services, example online education and business. This rapid increase in demand increased the burden on cloud services. Considering the fact that Covid-19 was not expected, the cloud was not optimised for this, at least not initially. Therefore it can be predicted that more failures should have occurred during Covid-19 period. As a result, it is also interesting to compare and provide analyses for the year 2020 (Covid-19 period) and the previous years.

1.5 Main results and contribution

[C1] A software to process raw cloud failure data and outputs useful information.

[C2] The first analyses of cloud failures prior and during Covid-19 period.

[R1] A report submitted for HP research project.

[R2] Cloud failure analysis that may be used to reduce cloud failures.

2

Background

The cloud failures data used by the study is obtained from the official website of the big cloud providers (6, 7, 8). The study covers the time period 2018 till 2020 (June). The failure information was initially put in rows and gathered in three separate files for each year named 'provider failures 2018', 'provider failures 2019' and 'provider failures 2020'. All three files had 12 columns that had information about: service id, service name, location, status, event start time, event end time, event duration hours, first notification, last notification, description, vendor, monitor and orgtype. Most column headers are self explanatory. The event start time, event end time, first notification and last notification column has information of date and time. This is originally represented in the form of a unix timestamp. The orgtype column means origin type. The origin type is 'cloud' for all rows as data is about cloud services. The study covers three big cloud services so the vendor column has three possibilities that are; AWS, Azure and GCP. The vendor column and monitor column contain the same information. For consistency, the vendor column was used when needed during the study. The program made to assist in cloud analysis was used to confirm the information mentioned about the columns. This was done by choosing the column and outputting unique values and their respective counts. Description of columns are show in table 2.1.

The file 'provider failures 2018' had 965 rows, file 'provider failures 2019' had 1024 rows and 'provider failures 2020' had 639 rows in total. The rows of 2020 file are less compared to other years files because 2020 covers cloud failures until June while other files have full year cloud failure information. The provided counts are of the raw data on which no operation had been performed. Each file contains information about three big cloud vendors AWS, Azure and GCP. In this initial data set rows were unorganised. The original

Column name	Description
service id	The id of the service
service name	Name of service
location	Region of failure
status	Indicators such as 1,0. No meaningful interpretation
event start time	Start time of failure
event end time	End time of failure
event duration hours	Derived feature representing duration of cloud failure
first notification	First notification of the failure issued by the vendor.
last notification	Last notification of the failure issued by the vendor.
description	Details of cloud failure.
vendor	The cloud provider AWS or GCP or Azure.
monitor	Same as vendor.
org type	For all instances the origin is cloud.

Table 2.1: Data columns description.

count for each vendor for each year file is given in the first row of the table, see Table 3.1. In general the data set contains rows with missing data and special cases. There are three different vendors and some do not provide information required to fill all 12 columns, see Section 3.1.2 . These cases are handled carefully at the right time in the right way and are explained at the respective steps, see Section 3.1.

The processes performed example duplicate removal, are the same for all three files because they have the same structure. At the start of the program, input is given to the program which is used to specify which year file should be processed. Input 1 to process file 'provider failures 2020', 2 for file 'provider failures 2019' and 3 for file 'provider failures 2018'. In any case the program reads/stores all three files. Reading all three files can enable comparing data in files, combined plots and other operations performed in later steps. The program prints the information to the screen. All operation results are verified by software and manual checks. Manual checks include observing different parts of data before and after the program operates. Multiple people are involved so error is minimized. After the checks, the program is improved when required. The study includes 2020 data till June. When 2020 is mentioned in this text, it means till 2020 June.

3

Results and Analysis

3.1 Cleaning Data

This section describes operations performed for cleaning data. The operations mentioned here are in order. If a process is mentioned before another process. It likely indicates that the process was done before the next process, unless specifically mentioned in the description. Therefore the understanding of the next process should be gained assuming the processes mentioned before are already completed.

3.1.1 Duplicates

The initial data set had duplicate rows, described in Section 2. Removing duplicates is the first step to prepare data. Duplicate rows are when a failure event is reported more than once in the file. Duplicate rows have exactly the same service_id, service_name, location, event_start_time and vendor.

Description is not included in the criteria for duplicate rows. Duplicate rows will have the same description because the event is the same. However, the string length can be different. For example there are two duplicate rows, the first row the description starts with "The" but the second row starts without the "The", the remaining description is the same for both rows. This example was observed during the study. We output the duplicate rows and study them manually to improve and verify the process of duplicate removal. Similarly we do not consider event end time when removing duplicates. There is a special case in which the rows are duplicate and have the same description length but have different end times. The quantity of such rows is very few. In this case we keep the row with higher end time. Comparing descriptions and event end times of two rows would not be beneficial as seen in the example. Instead of using description we use other fields

(mentioned above) to check for duplicates. The program does the following to remove duplicates: the program starts at the first row, selects a row and compares with the rows below the selected row in the file. If a duplicate row is found, it is removed from the file. In normal cases, when two rows are exactly the same, the selected row is kept and the other rows are deleted from the file. The table 3.1 shows the result of removing duplicates.

Number of	2018				2019				2020			
Rows	**AWS**	**Azure**	**GCP**	**Total**	**AWS**	**Azure**	**GCP**	**Total**	**AWS**	**Azure**	**GCP**	**Total**
Initial	325	284	355	964 +1	273	222	528	1024 +1	271	51	316	638 +1
Duplicate	4	145	142	291	7	110	261	378	1	15	114	130
After Removing Duplicates	321	139	213	673 +1	266	112	267	645 +1	270	36	202	507 +1

Table 3.1: Results of duplicate removal.

The plus 1 in table means that there was 1 row that had no vendor information. In both 2019 and 2018 files, this row gets removed at a later step, see Section 3.1.3.

3.1.2 Unknown cells and data limitations

The unknown cells include the cells that are empty cells or cells with '-1' or '0'. For consistency we aim to replace the various representations and use 'Unknown' for string based unknown cells and 0 for integral based unknown cells. After doing the process of different representation and duplicate removal, it is observed when doing this process, there are no unusual unknown rows, some patterns are clearly observed. The reason is that not all vendors provide all information as addressed by the current study, see Section 2. The unknown cells are GCP location cells in 2020 and 2019 data files, see Section 3.1.3. Another case is Azure does not have service id in any row in any year. Azure also does not provide information about first notification and last notification. These are also the limitations of the data and the study. All other rows are filled with appropriate information. Table 3.2 gives the frequency of the unknown cells as per vendor and total.

3.1.3 Incorrect dates

In the initial files the files had rows that had incorrect date and time. The rows had "-1" as event start time and event end time. "-1" indicates that event time was not known. Due to lack of information, the rows with "-1" were removed. There were also rows that

had event time of the previous year, for example the file of 2020 contained failure events of 2019 and 2018. The old dated events were already present in the correct year files and were repeated in the new year file. The old dated rows were not required in the new year file so they were removed from the file. The results are shown in table below.

Number of Rows	2018	2019	2020
After Removing Duplicates	674	646	509
Incorrect dated Rows	275	271	386
After removing incorrect dated rows	399	375	123

Table 3.2: Missing dates and old dated rows.

3.1.4 Special case rows

The initial files also had other special cases related to event start, end date and time. Few rows had the same event start and end time. Same event start and end time is not possible so these rows are considered incorrect and removed. Another case is that some rows had event end time before the event start time. These rows were studied manually and checked with the time in the description. A common thing that was noticed in rows having end time before start time was that in all cases the start time was not converted to a 24 hour format. We converted these to 24 hour time and the times matched the timing mentioned in the description. In general these rare cases were present in only a few rows. In total less than 15 rows per file had end time before start time. In this case rows were kept in the files fixed by converting to 24 hour format. Another case was that there were few rows that had corrupted description. These rows count at most 5. Corrupted description means the rows description had symbols not understandable English. These rows were removed.

After taking care of the special cases, the frequency distribution of each file is shown in table 3.3.

3.1.5 Different words representing same meaning

After cleaning the data and sorting rows, the data was more readable. The quantity was also reduced compared to initial files. At this stage data files were manually observed.

Number of Rows	2018	2019	2020
After removing incorrect dated rows	399	375	123
Special cases rows	13	18	9
Remaining rows	386	357	114

Table 3.3: Result of fixing incorrect date and special cases.

It was noticed that in the column of location and service name, some rows needed to be repaired. Two rows have location 'East US' but the second row uses lowercase 'east us'. It is likely that when performing comparisons between two cells, this can give incorrect results. Another scenario is that 'Networking' and 'Network' represent the same service name. Similarly 'Southeast Asia' and 'South East Asia' represent the same location. When two cells intend to deliver the same information, we replace them with a single representation, for example we replace 'Southeast Asia' and any other representation of southeast asia with 'South East Asia'. These measures are helpful in reading and sorting data. It is important for filtering because this makes a single representation and provides the correct result when filtering an example when outputting unique values. This operation is done before removing duplicate rows, this improves the result of the duplicate removal process. The information, for example the frequency count presented in the current study is the result of after doing this process.

Number of	2018				2019				2020			
Rows	AWS	Azure	GCP	Total	AWS	Azure	GCP	Total	AWS	Azure	GCP	Total
Initial	325	284	355	964 +1	273	222	528	1024 +1	271	51	316	638 +1
Total Removed	188	165	255	578 +1	155	144	367	667 +1	233	34	257	524 +1
Remaining after cleaning	137	119	130	386	118	78	161	357	38	17	59	114

Table 3.4: Frequencies of data removed during cleaning process.

3.1.6 Organising Data

The data rows in the initial files were disorganized[point to background section]. Generally there were many sorting options. In particular, the order location, vendor, service id, event start time, and then service name was found to be the optimal sorting order. Rows are sorted by location first, then vendor, and so on. The information was arranged in ascending order. The sorting operation was very useful in certain analyses and also in outputting data.

The data has been cleaned and organized into groups. We have the data as separate year data files initially. We store data after it has been cleaned with respect to three different vendors for each year. This can be useful for analyses and in other operations, such as graph creation.

3.2 Analysis

3.2.1 Monthly Plot

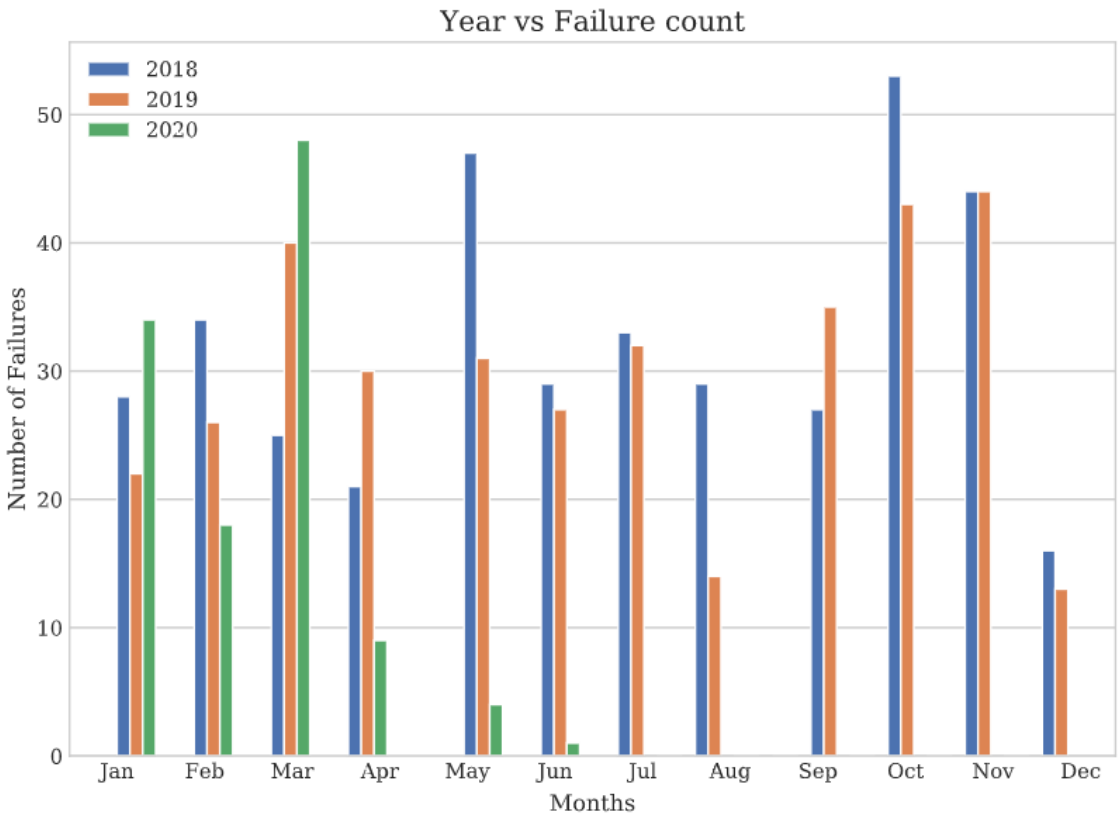

Figure 3.1: Plot of failure frequency in each month.

O-1: The months of October and November had the greatest failure frequency, while December had the lowest number of cloud failures over the years.

The figure 3.1 shows the total number of monthly cloud failures in 2019 is less than in 2018 except for March, April, and September (2020 not included). That is, in 75 percent of cases the cloud failures have decreased. In both 2018 and 2019, October and November are 'peak' months where cloud failures are relatively high compared to the rest of the year. During these months cloud failures reach very close to the highest failure count of the year.

In December the cloud failures are the least for both 2018 and 2019. Cloud failures for March have been increasing every year, while in February and May cloud failures have decreased continuously during the three years. The high peak in March 2020 is likely

related to Covid-19 the evidence being that this is the first month since October 2018 that cloud failures reach such height (approximately after 1.5 years). This month was the peak lockdown month and most works had been shifted online. The sudden high usage of cloud services during March due to Covid-19 can be a possible explanation for the high peak.

O-2: Failure peak in March 2020 related.

An interesting point is that May 2020 had the least number of cloud failures in three years while March 2020 was among the highest cloud failure month. Overall, the pattern of bars shows that when failures are high in a particular month, they do not continue to remain high in upcoming months (high referring to more than 30 failures). The bars always come down in the next month.

3.2.2 Comparing Vendors

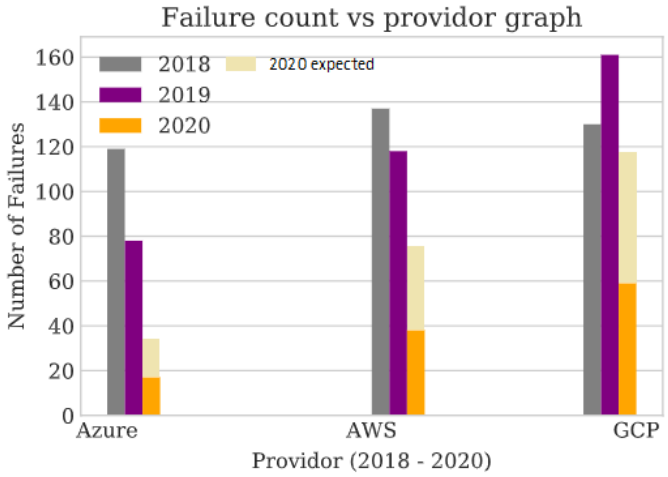

Figure 3.2: Failure count in cloud service providers.

O-3: Microsoft Azure had fewer failures than AWS and GCP.

In 2018 AWS had the most cloud failures, in 2019 GCP had the most cloud failures and in 2020 GCP is expected to have the most failures. Plot 3.2 shows that in these three years Azure had fewer failures than AWS and GCP. For both AWS and Azure number of failures decreased from 2018 to 2019. However, the number of failures for GCP increased from 2018 to 2019. Overall, during the 2 and half year Azure has smaller failure count bars compared to other vendors.

3.2.3 Failure time and recovery

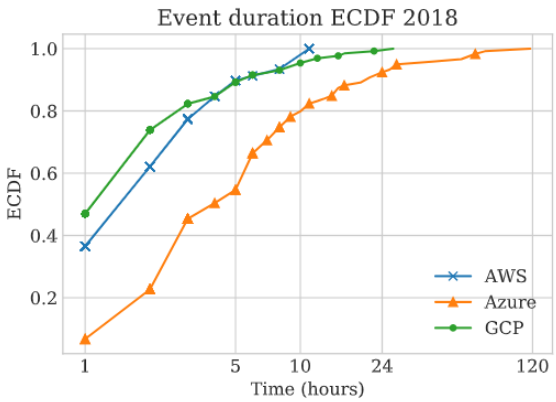

Figure 3.3: ECDF pLots of 2020.

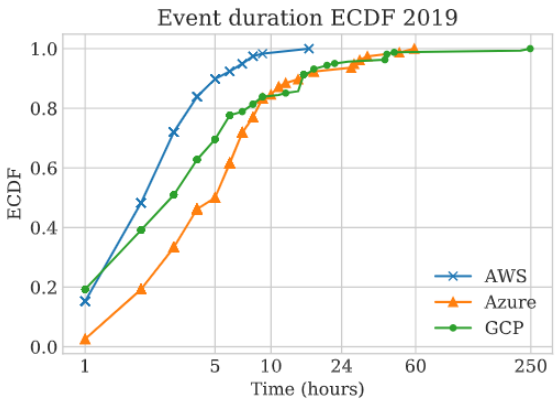

Figure 3.4: ECDF pLots of 2019.

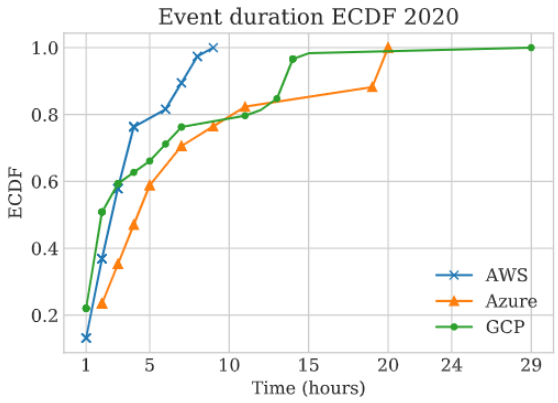

Figure 3.5: ECDF pLots of 2018.

O-4: AWS had the fastest recovery, followed by GCP. Microsoft Azure recovery was more delayed compared to the other vendors. For all vendors approximately 50% of cloud failures were recovered in less than 5 hours.

Form the plots in figure 3.3, 3.4 and 3.5 it can be observed that AWS and GCP recover 70 to 90 percent of failure events in 5 hours while Azure recovers only 50 to 60 percent. In general, during the 2 and half year time, AWS has the quickest recovery of cloud failures. GCP recovery speed although behind but is close to AWS. For Azure, the failure event recovery is slower compared to other two vendors.

Overall, AWS outages have lasted no longer than 24 hours. GCP had the longest failure occurrence in 2019, spanning up to ten days. In particular, the longest failure occurrence

in 2018 was of Azure, which lasted 120 hours. The longest failure occurrence in the first half of 2020 was of GCP, which lasted 29 hours. Eighty percent of cloud outages were recovered within ten hours for all three vendors. In addition, half of the failure events lasted fewer than 5 hours.

3.2.4 Cloud Services Failures

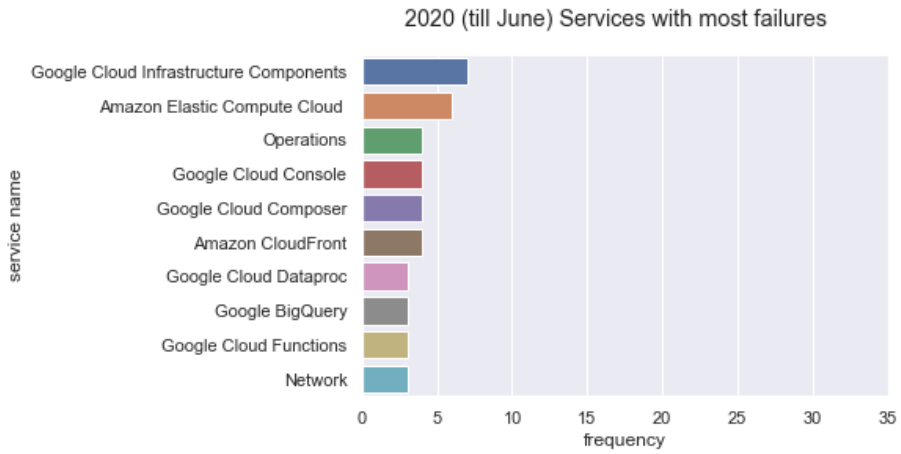

Figure 3.6: 2020 services with most failures.

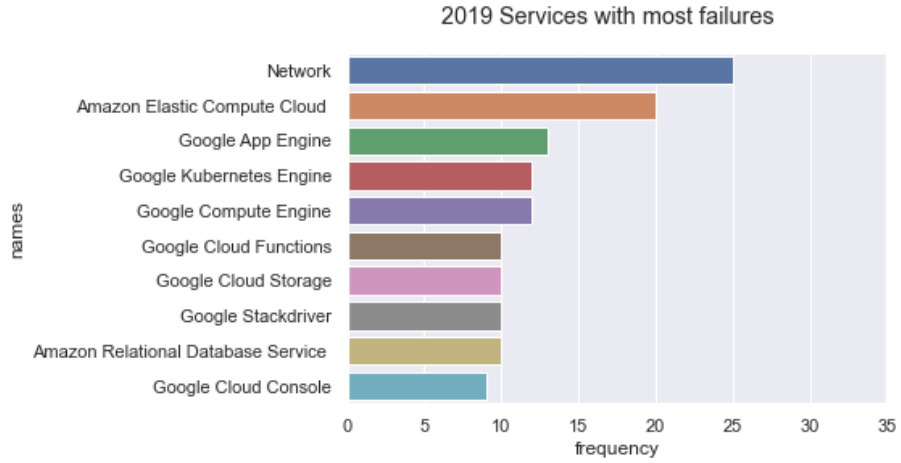

Figure 3.7: 2019 services with most failures.

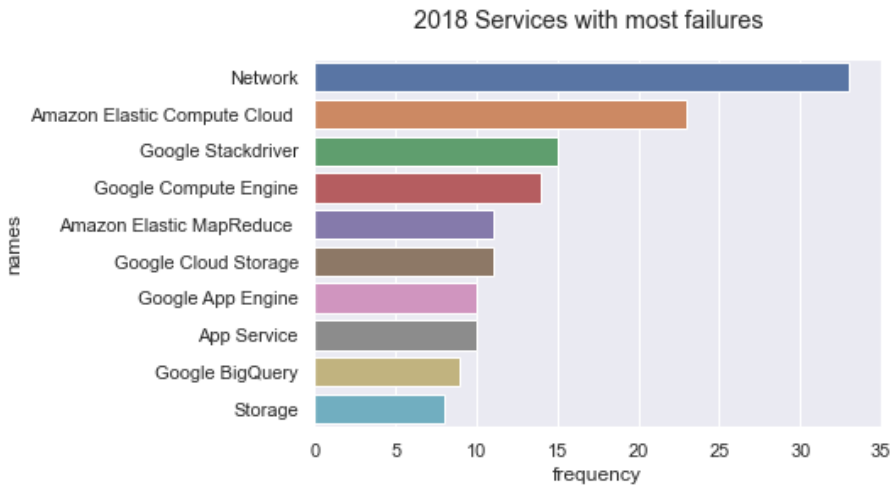

Figure 3.8: 2018 services with most failures.

O-5: Network service was among the frequently failing service and in total had the most failures over the time period. There are many services with continuing failures over the years.

Figures 3.6, 3.8 and 3.8 shows the name and frequency of the ten services that had most failures in 2020, 2019 and 2018 respectively. A common services failing in all three years is Network service, this is the most failing service throughout the 2.5 year and this service also has the highest number of failures in 2018 and 2019. Similarly Amazon Elastic Compute Cloud service is the second most failing service in all 2.5 years.

Services that have been failing for exactly two consecutive year's (2018 and 2019) include Google Cloud Storage service, Google Stackdriver, Google App Engine and Google Compute Engine. In most cases the number of failures of these services have decreased compared to previous year. Services that have been failing for exactly two consecutive year's (2019 and 2020) include Google Cloud Functions and Google Cloud Console.

3.2.5 Time of Cloud Failures

According to weeks in year

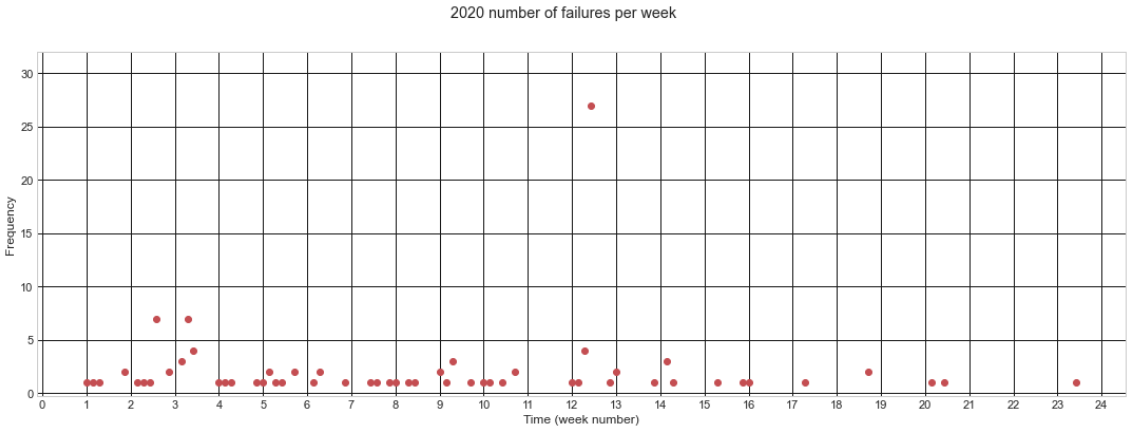

Figure 3.9: 2020 Failures till end of June.

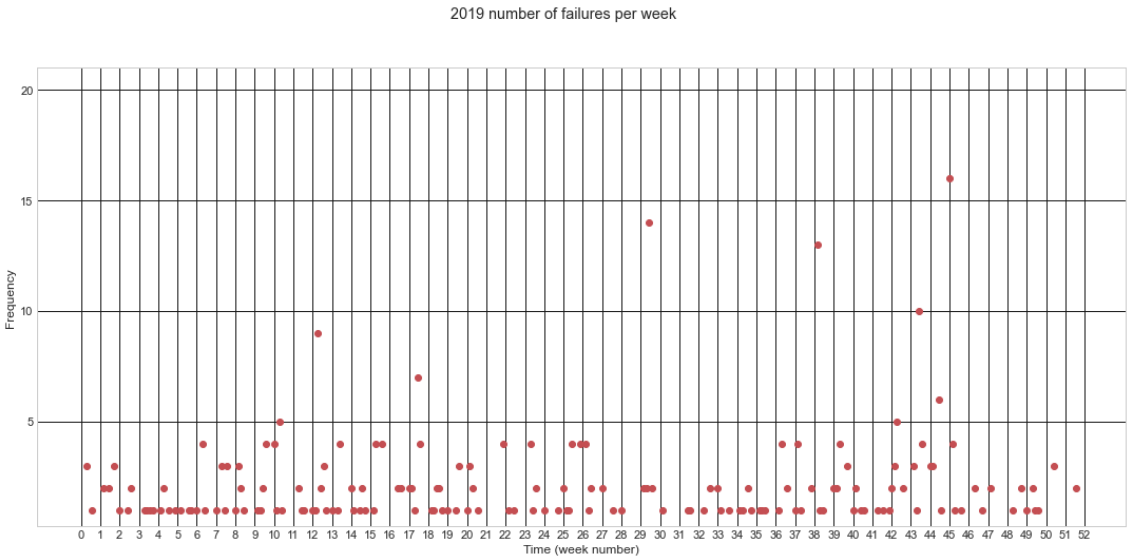

Figure 3.10: 2019 Failures throughout year.

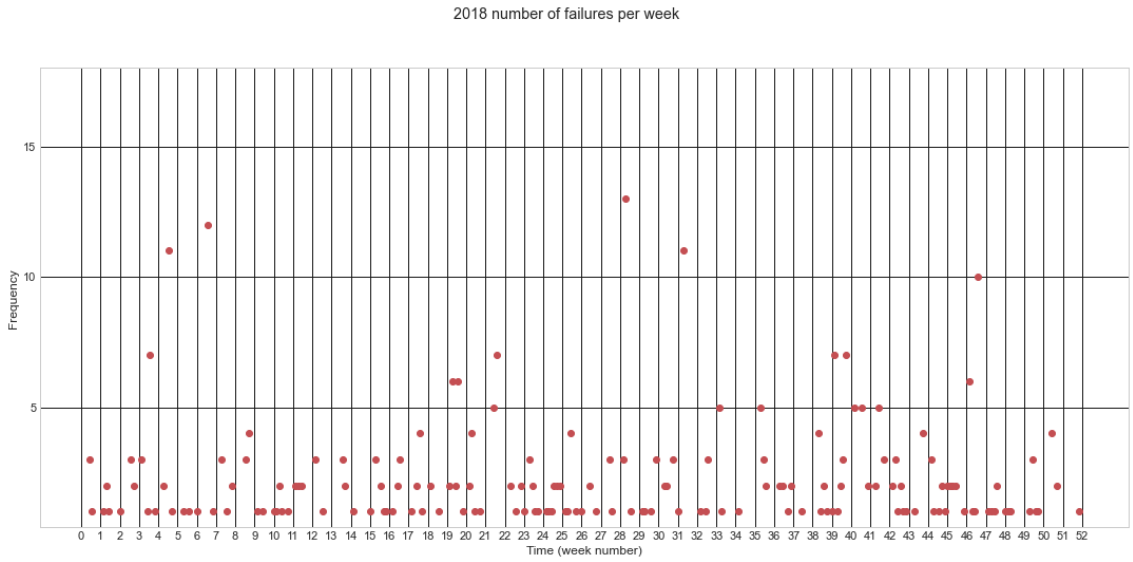

Figure 3.11: 2018 Failures throughout year.

O-6: More than 25 services failed on a day in 2020. Furthermore, a week without cloud failure is rare.

Figures 3.9, 3.11 and 3.11 shows the failures during the weeks of the year 2020, 2019 and 2018 respectively. The horizontal axis shows the week number while the red dot shows the failure at the day of the week. In 2020, the highest red dot in week 13 can be clearly noticed indicating failure of more than 25 services on a day. In 2019, more than 10 failures on a day in week 30, 39 and week 46 with highest that is more than 15 failures on the day. Similarly in 2018, more than 10 failures in week 5, 7, 33 and 29 with highest failures. In all three plots we can notice that it is rare to have a week without failures which is a proof of importance of this study. From 2018 to 2019 failures are present but in 2019 failures on a day are not high such as majority less than 5 failures, as a result that is 2019 plot is less scattered vertically.

According to days of week

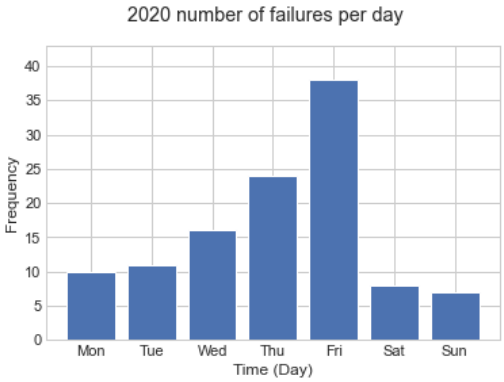

Figure 3.12: 2020 Failures in week days.

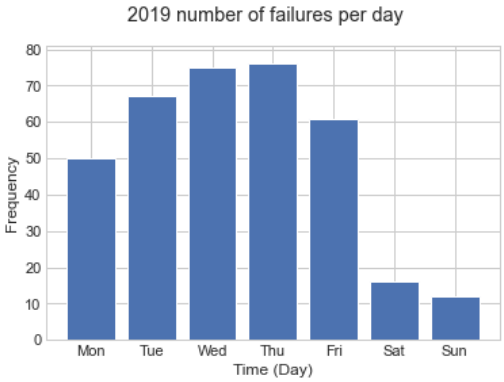

Figure 3.13: 2019 Failures in week days.

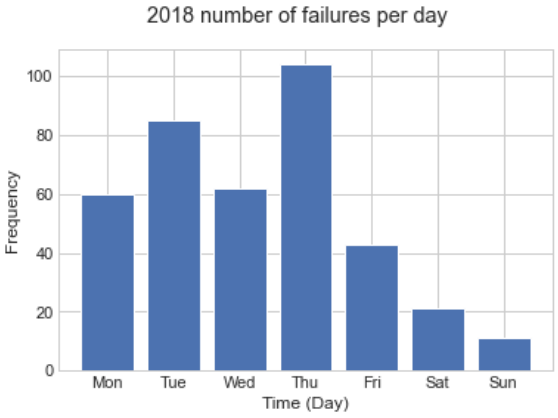

Figure 3.14: 2018 Failures in week days.

O-7: Highest cloud failures around Thursday and Friday while lowest on Sunday.

Figure 3.12, figure 3.13 and figure 3.14 shows the failures at days of the week in 2020, 2019 and 2018 respectively. In 2018 and 2019 we can observe and it is also expected that more failure occur during working days (Monday to Friday) than on weekend (Saturday and Sunday). In 2020 the pattern is present but may not be clear as only half year data. For both 2018 and 2019 Thursday has the highest number of failures. While in 2020 Friday has the highest number of failures followed by Thursday. Throughout the 2.5 years Sunday has the least number of failures

3.2.6 Location of Cloud failures

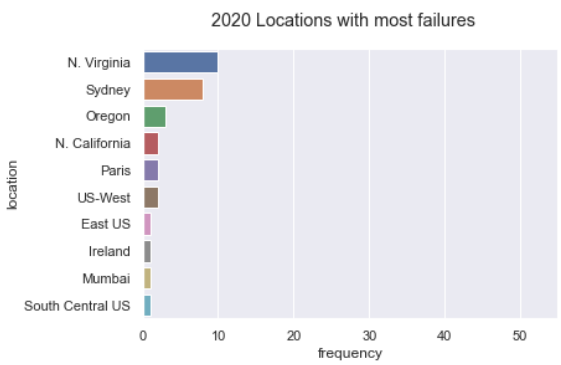

Figure 3.15: 2020 locations with most failures.

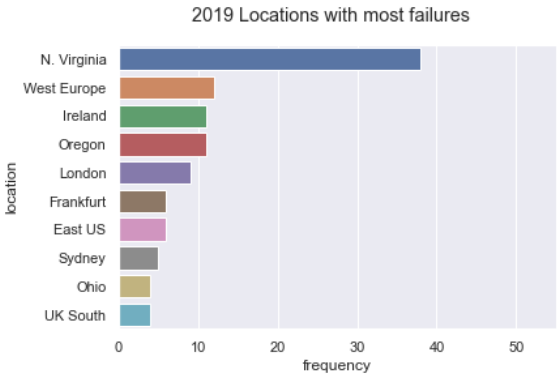

Figure 3.16: 2019 locations with most failures.

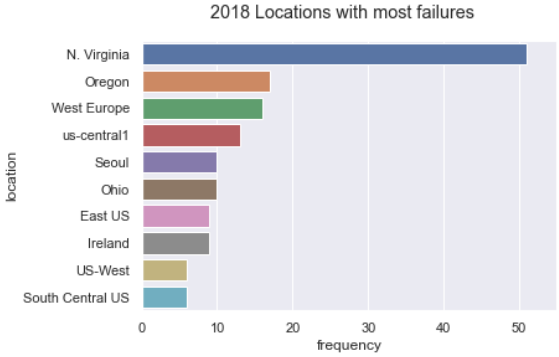

Figure 3.17: 2018 locations with most failures.

O-8: North Virginia and Oregon have the highest and most frequent cloud failures.

Figure 3.15, figure 3.16 and figure 3.17 shows the name and frequency of the ten locations that had most failures in 2020, 2019 and 2018 respectively. The location with highest number of failures during the 2.5 years is North Virginia reach 51 failures in 2018 which decrease to 38 in 2019. Another common location throughout 2.5 years is Oregon having above 10 failures during 2018 and 2019 (2020 data is only of half year). Other common locations are East US, Ireland and South US.

The location analysis is limited to the data available that is we do not know locations of all failures. In plots and analysis we also ignore the failure that have multiple origins.

4

Evaluation and Related Work

4.1 Analysis of Limitations

The research looks at three different providers. The vendors do not give all of the data for detailed study. This is one of the limitations from the data that is the results depend on the accuracy of the data available see section 3.1.2. Another restriction is that the study only includes data from January 2018 to June 2020 about two and a half years. When striving for definite conclusions from the data, this quantity of data can be deemed restricted; for example, data from more prior years can improve data comparison.

The study's validity is maintained by using the software and confirming the intended behaviour by performing the same operations on multiple files. Furthermore, the procedure was carried out in a variety of ways to ensure that the results produced were consistent. Manual inspections were also carried out by examining the program outputs and comparing them to the expected (manual) output. Moreover, majority of operations are automated, which reduces the risk of human error.

4.2 Related Work

Currently, only a limited amount of work has attempted to analyze cloud failures of big vendors. For example, the authors of (9) analyze outages and incidents reported by companies and news outlets. The current study's data was gathered from official sources (6, 7, 8) of the vendors. Furthermore, the current study is one of the first because it is based on recent time periods. There are studies that aims to determine the possible causes of cloud failure using failure analysis (1) using statistics from newspaper articles to analyse cloud

failures. However, no study exists that examines the failures on the cloud of the three major cloud service providers, AWS, Azure, and GCP.

5

Conclusion

5.1 Conclusion and Future Work

Cloud services are beneficial to people all around the world. The cloud is viewed as a solution to a variety of issues. For example, the cloud allows people to collaborate and communicate with one another, particularly when they are in different countries. There is, however, a long list of cloud failures that could have a detrimental impact on billions of cloud users. Understanding cloud failures is essential. We created a tool to help with the process, and then then analyze and understand cloud failures. By studying cloud failures, many of cloud failures can be prevented along with the loss caused by cloud failures. This study covered the period 2018 till 2020-June and provided analyses of cloud failures in big cloud providing companies; AWS, Microsoft Azure and GCP. For further study, the tool can be extended to be used for analysis for future cloud failures. Furthermore, this study can provide insight for reasons of cloud failures occurance.

6

Self-Reflection

6.1 Self-Reflection

Through this project I learned about growing demand and importance of cloud services. It was interesting to learn about cloud failures including why, where, how they occur and their possible solutions. During the research I interacted with many professional members of the team that guided me through out the study. I was greatly inspired by their words that introduced my to methodologies that I had not known. The research showed me a variety the techniques to analyse data. This study boosted my skills as a python programmer as I gained experience by creating a software for the study. I learned programming skills used for data analysis, applying statistical methods, plotting graphs using code, various ways of representing data and different types of graphs such as the ECDF plot. The most attractive part of the study was the results of analysis.

During the research two-third of the time was spent on data cleaning and visualization. Among which more time was spent by cleaning data (RQ1), than on data transformation (RQ2). The data provided was raw data. Converting the raw data in use-able data was done using a software created during the study. Furthermore cleaning data involved many operations, see 3.1 The leftover one-third time was spent on analysis and documentation.

References

[1] Haryadi S. Gunawi, Mingzhe Hao, Riza O. Suminto, Agung Laksono, Anang D. Satria, Jeffry Adityatama, and Kurnia J. Eliazar. **Why Does the Cloud Stop Computing? Lessons from Hundreds of Service Outages**. In Marcos K. Aguilera, Brian Cooper, and Yanlei Diao, editors, *Proceedings of the Seventh ACM Symposium on Cloud Computing, Santa Clara, CA, USA, October 5-7, 2016*, pages 1–16. ACM, 2016. ii, 22

[2] Haryadi Gunawi, Vincentius Martin, Desmond Anang, Mingzhe Hao, Tanakorn Leesatapornwongsa, Tiratat Patana-anake, Thanh Do, Jeffry Adityatama, Kurnia Eliazar, Agung Laksono, and Jeffrey Lukman. **What Bugs Live in the Cloud?** pages 1–14, 11 2014. ii

[3] Katsantonis Konstantinos, Persefoni Mitropoulou, Evangelia Filiopoulou, Christos Michalakelis, and Mara Nikolaidou. **Cloud computing and economic growth**. 10 2015. 1

[4] Gartner Inc. **Gartner forecasts worldwide public cloud revenue to grow 17% in 2020**. 10 2019. 1

[5] Ana Gainaru, Franck Cappello, Marc Snir, and William Kramer. **Fault prediction under the microscope: A closer look into HPC systems**. In *SC '12: Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, pages 1–11, 2012. 2

[6] Microsoft Azure. Azure Status History. https://status.azure.com/en-us/status/history/. Accessed: 2020-04-15. 6, 22

[7] Google Cloud Platform. Google Cloud Incidents JSON Feed. https://status.cloud.google.com/incidents.json. Accessed: 2020-04-15. 6, 22

[8] AWS. AWS Status JSON Feed. http://status.aws.amazon.com/data.json. Accessed: 2020-04-15. 6, 22

[9] Lance Fiondella, Swapna S. Gokhale, and Veena B. Mendiratta. **Cloud Incident Data: An Empirical Analysis**. In *2013 IEEE International Conference on Cloud Engineering (IC2E)*, pages 241–249, 2013. 22