Vrije Universiteit Amsterdam

Honours Programme, Project Report

This is a title

Author: Alexander Balgavy (2619644)

1st supervisor: Alexandru Iosup
daily supervisor: Sacheendra Talluri

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.

June 1, 2020

Abstract

This is an abstract.

1 Introduction

Cloud computing as a paradigm has become a de-facto standard for services and applications. We build our modern distributed services in an increasingly 'cloud-native' manner, based on cloud-computing abstractions (IaaS, SaaS, etc.) and technologies (VMs, containers, functions, etc.). These 'cloud services' can be client-facing (such as the Netflix web application), or on the back-end (e.g. Amazon Simple Storage Service). In fact, many organisations that are essential for society (such as banking or healthcare) rely on cloud services which are invisible to end users [1], [2]. But how reliable are these services?

Though cloud services are expected to always be available, they can fail in a variety of ways. These failures can range from mild, such as a single-region outage affecting some users, to severe, such as a multi-region outage of multiple services. Client-facing services are prone to upstream errors, and can fail when any one of their dependencies fails [3]. Such a failure results service downtime, which can lead to disruption of critical services [4], [5], data loss [6], and other issues.

Therefore, it is important to understand how cloud services fail. If we can understand the characteristics of these failures, we can potentially discern reasons for the failures. This can aid cloud providers in the prevention of outages, and allow them to further increase the availability and fault-tolerance of their services.

The main contribution of this study is a systematic analysis of provider-reported failures during one calendar year. We develop a framework for conducting such an analysis, which can be used in future studies. For me, the main contribution of this research was learning how to process and clean a dataset, and how to iteratively develop a methodology to analyze the dataset.

2 Background information

3 Methodology

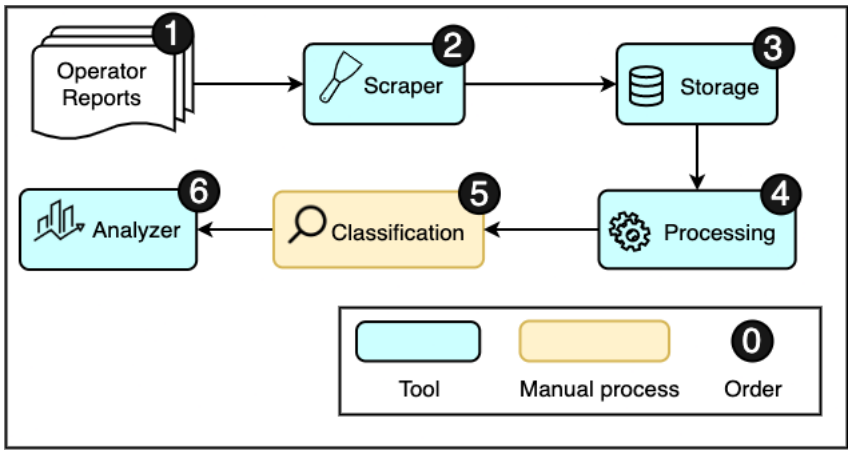

Figure 1: Data collection, extraction, and analysis process.

Cloud providers self-report information about service availability and failures through public status pages. We selected three of the largest worldwide cloud service providers: Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform (GCP). The whole workflow is shown in Figure 1. We use data scraped from their public status pages for self-reported failures [7]–[9] in the span of one year, Jan-Dec 2018. The pages were scraped every six hours, to avoid burdening the

page host with more frequent scrapes and potentially incurring penalties (step 2 in Figure 1). The raw dataset is approximately 522 KiB in size; the cleaned and processed dataset is around 172 KiB.

We use Python 3.7.7, Pandas 1.0.3, NumPy 1.18.4, SciPy 1.4.1, and Pytz 2019.3 to process and analyze the extracted structured failure data. We first deduplicate the data based on the service name, failure location, and event start time, selecting the events with the longest duration (step 4 in Figure 1). This removes 400 events in total: 137 from Azure, 127 from AWS, and 136 from GCP. We convert all reported event start times to the timezones appropriate to the region specified for the outage. We calculate the event duration in minutes based on the event start and end times. We also compute the year the event took place, as well as the hour of week (with hour 0 denoting midnight on Monday). We then filter the data, selecting events that occurred in the 2018 calendar year. This yields a grand total of 411 events: 139 from GCP, 144 from AWS, and 128 from Azure.

The providers also include in their reports a textual description of the outage. The description does not follow a specific format or standard, and must thus be manually analyzed for each event. We classify outages across several categories based on the description of the outage (step 5 in Figure 1). The classification is partially based on that done by Gunawi et al. [10], [11], but as many of the failure descriptions are terse and brief, our classification is less detailed.

The description provides information about the qualitative aspects of the outage. We identify seven such aspects: how many services were affected, the severity, the range, the users affected in the outage, the root cause of the outage, the duration of the outage, and the components of the service affected in the outage. An outage can affect one or multiple services. The severity can be visual (i.e. performance is not affected, only visual feedback is incorrect), degradation of performance, or complete unavailability of the service. The range of an outage can be (in ascending order by geographical size) a single availability zone, a single region, or multiple regions. As the range of a single availability zone only applies to AWS services, outages that occurred in a single availability zone (13 events) were merged with those occurring in a single region, to simplify analysis. An outage can affect some users, or all users; this is meant to be understood in combination with the range of the outage (e.g. an outage may affect all users in a single region). The cause of an outage can be a code error, a side effect of maintenance, a configuration error, a network error, an external factor, increased load, or an unhealthy unit. A "unit" is not necessarily a physical (i.e. hardware) unit, but can be a virtual node, a cluster, etc.

Each of these causes can be further separated into narrower categories. A configuration error can stem from a direct change to a configuration file, or can happen as part of a deployment task. A network error can indicate an internal API issue, or an internal network issue. An external factor can be the environmental conditions in the datacenter (e.g. higher temperatures or humidity), a shock event (e.g. a thunderstorm), or an issue caused by a third party. If one of these features is not stated in the description of a particular outage, we note that it was not provided for the outage.

4 Results & analysis

Hourly and daily failure trends We first observe the distribution of outages across the week, this is shown in Figure 2. GCP and AWS both display significant peaks at two points during the week: around the middle (Tuesday and Wednesday), and at the start of the weekend (Friday and Saturday). AWS also shows an increase in outages in the afternoon/evening hours on Sunday. The data provided by Azure does not indicate any clear trends.

Root causes of outages by impact level and vendor We next analyse the distribution of outages across root causes and impact levels. We define the impact level of an outage as the combination of the number of services affected and the geographical range of the outage. We identify seven classes of root causes: UNIT (individual nodes, instances, or clusters, not necessarily hardware), NETW (related to the internal or external network), MAINT (side effects caused by maintenance), LOAD (increased load on the service), EXTERN (external causes, i.e. environmental or third-party), CODE

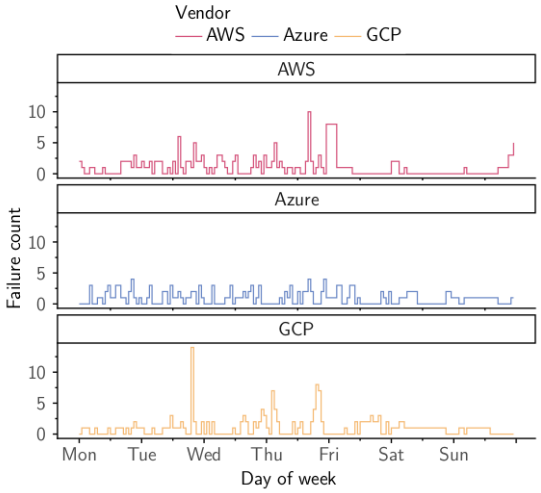

Figure 2: Distribution of outages across the week, by vendor.

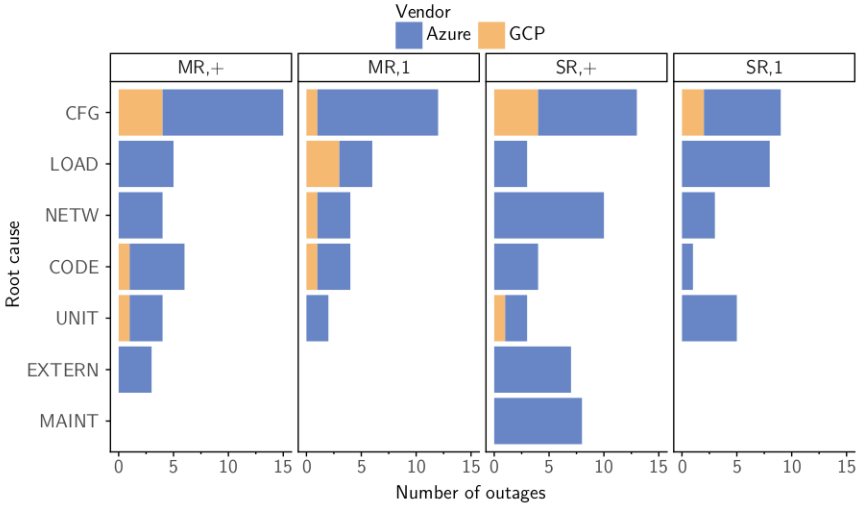

Figure 3: Root causes of outages across varying impact levels, by vendor. (MR = multiple regions, SR = single region, 1 = one service, + = multiple services)

(code errors/bugs), and CFG (configuration errors). We further separate the outages by vendor, indicated by the color of the bar in Figure 3. We do not include AWS, as we do not have sufficient data regarding root causes from AWS (a cause was only specified for 2.1% of AWS events). Excluded from the plot are outages that did not provide a root cause (a total of 122 events, 45.69%), a range (5 events, 1.87%), or the number of affected services (4 events, 1.5%).

The majority of outages across all levels of impact are caused by configuration errors. For multi-regional outages, the configuration category accounts for the majority by a wide margin. On the other hand, for single-region outages, there are multiple leading causes: apart from configuration errors, outages affecting one service are also caused by increased load and failing instances, and those affecting more than one service are also caused by network errors and maintenance side effects.

Failure distribution across the week, separated by root cause We also analyze the frequency of root causes at various points in the week, separated by vendor. We do not include AWS outages here, due to a lack of specified root causes. Outages that did not provide a root cause are also excluded (262 events, 63.75%). From Figure 4, it seems that load-related outages tend to happen starting between midday on Wednesday and Thursday evening, with smaller peaks during the night on other days. Most outages caused by misconfiguration happen in the first half of the week, from mid-Monday until midnight on Thursday. This could perhaps be because large code changes

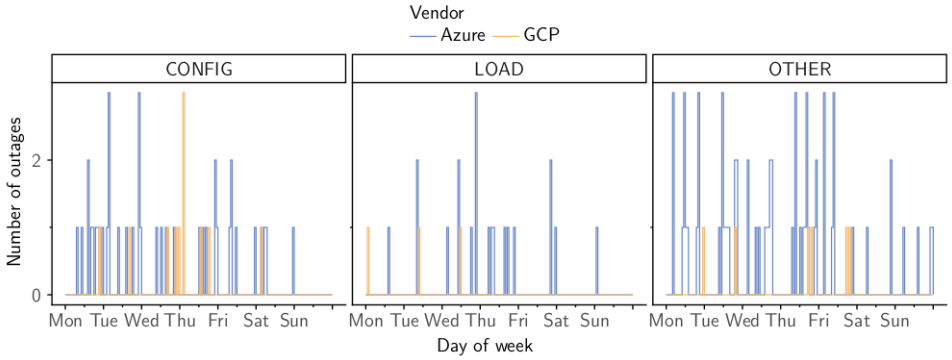

Figure 4: Outages across the week, separated by root cause and vendor. (CONFIG = configuration error, LOAD = increased load, OTHER = other)

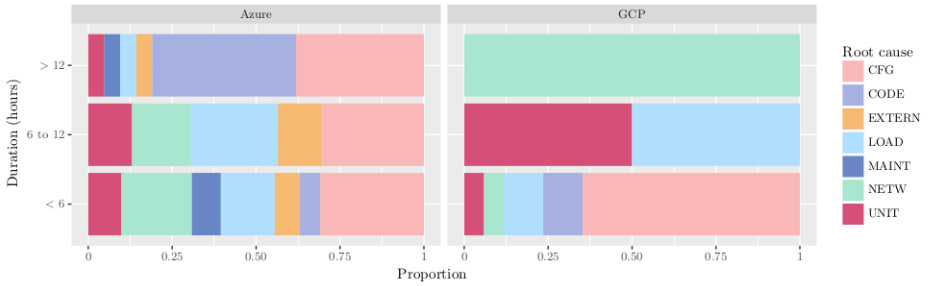

Figure 5: Root causes of outages, separated by duration and vendor.

are usually introduced during the first few days of the week. The major peaks for these outages generally occur around midnight. The reason for this could be that deployments happen during the night, when it is likely that fewer customers will be using the service; as Langford et al. found [12], there is a significant difference in traffic during the day and during the night. It could also be that changes are deployed during the day, but there is a delay before bugs appear. Outages related to other root causes generally occur during the weekdays, and there are only a few peaks during the weekend. It is important to note here that this is a relatively small dataset, and the trends observed in Figure 4 could change if more data becomes available.

Root causes of outages, by duration and vendor We separate the outages by duration, in three categories: short (outages lasting less than six hours, '< 6'), medium (six to twelve hours '6 to 12'), and long (more than twelve hours '> 12'). The outages are grouped by duration and vendor, such that the horizontal axis shows the proportion of outages for a specific duration category and vendor. We do not plot outages that did not specify a root cause (262 events, 63.75%). All AWS outages that specified a cause were shorter than six hours, and were caused by external events (e.g. third party or environment). However, as discussed above, only three AWS outages specified a cause, so AWS is not included in Figure 5. Azure services have more diverse outage causes than GCP services. All outages of GCP services that lasted longer than twelve hours were caused by network errors; in contrast, network errors did not play a role in these types of outages for Azure services. The longest outages of Azure services were caused in approximately equal parts by code errors, and by configuration errors. Configuration errors were a common cause of Azure outages for all three duration categories, while only the shortest GCP outages (shorter than six hours) had configuration errors as a cause. For both GCP and Azure, increased load was a main factor in the short- and medium-duration outages. For GCP, there were more medium-duration outages caused by increased load than short-duration outages; for Azure services, the proportion is approximately the same. GCP also had failing computational units as a major cause for medium-duration outages; this was not as prominent for Azure services, where failing units accounted for approximately the same proportion of short-duration outages as for medium-duration outages.

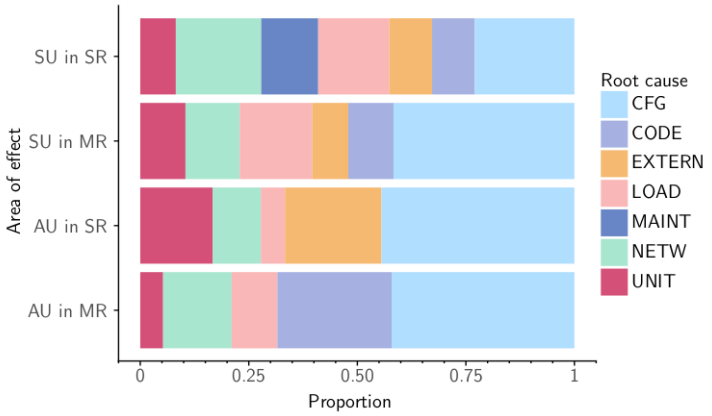

Figure 6: Root causes of outages at various areas of effect. (SU = some users, AU = all users, SR = single region, MR = multiple regions)

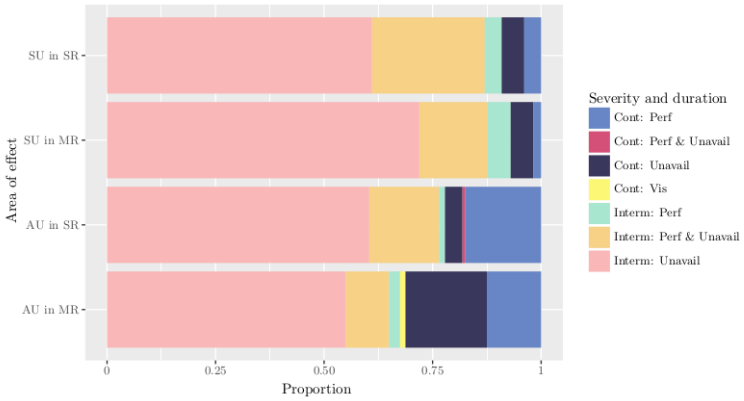

Figure 7: Severity of an outage and the users affected by the outage. (SU = some users, AU = all users, Cont = continuous, Interm = intermittent, Perf = performance degradation, Unavail = unavailable, Vis = visual)

Root causes of outages, by area of effect We also analyze the root causes of outages, separated by the area of effect. We define the area of effect of an outage as the affected users (all or some) and the range (single region or multiple regions). Here we exclude those events that did not provide a cause (262 events, 63.75%), a range (5 events, 1.22%), or the affected users (1 events, 0.24%). From Figure 6, we conclude that configuration errors account for the majority of outages with the widest area of effect (all users in multiple regions). The second main cause of such outages are code errors, which interestingly do not play a major role in single-region outages, and do not appear as a cause of single-region outages affecting all users. Failing instances are mainly a cause of single-region outages, more so for outages that affect all users – this is probably due to the fact that individual instances, nodes, or clusters are localised in a single region [13]. From the available data, it seems that outages caused as a maintenance side effect only affect some users of the service, mostly in a single region. It also appears that outages caused by increased load more commonly affect only some users in a given range. The majority of outages caused by network issues affect some users in a single region, though they also play a somewhat significant role in the other area of effect categories (accounting for around 10% of the outages in each category).

Severity and duration of failures by the area of effect Next, we consider the severity and duration of an outage, depending on its area of effect. We exclude events that did not state the affected users (1 events, 0.24%), range (5 events, 1.22%), severity (41 events, 9.98%), or duration (41 events, 9.98%). The first immediate finding from Figure 7 is that the majority of outages result in one or more services becoming intermittently unavailable. For outages affecting all users, in a single region or in multiple regions, there is a higher proportion of outages that cause a service to

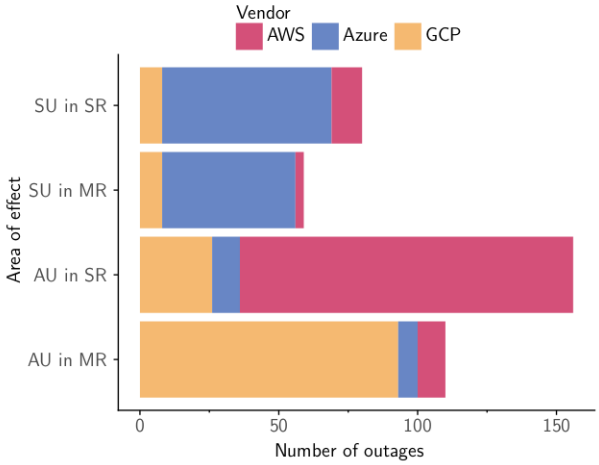

Figure 8: Area of effect of outages by vendor. (AU = all users, SU = some users, SR = single region, MR = multiple regions)

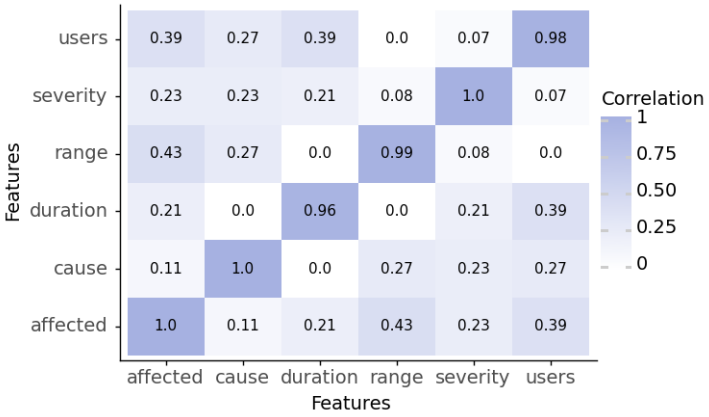

Figure 9: Correlation matrix (Cramér's V)

be intermittently degraded or unavailable than for outages affecting some users. Furthermore, the majority of outages that resulted in a service being continuously unavailable (for some period of time) affected all users in multiple regions. When the performance of one or more services is degraded, it generally happens for a continuous period of time.

Outages separated by area of effect and vendor We continue with the definition of the area of effect as the affected users and the range, and analyse the distribution of outages per vendor, in figure Figure 8. We do not include events that are missing information about the users (1 events, 0.24%) or the area of effect (5 events, 1.22%). We observe that the range of the majority of outages depends on the vendor. The majority of GCP outages affect all users in multiple regions, while the majority of AWS outages affect all users in a single region. For Azure services, the outages mostly affect some users, with an approximately equal distribution between a single region and multiple regions.

Correlation analysis (Cramér's V) Finally, we conduct correlation analysis on the various features of the data. The first statistic we consider is Cramér's V (also known as Cramér's phi coefficient, written as ϕ_c), which measures association between two nominal variables. It extends the phi coefficient to contingency tables larger than 2×2, and results in a value between 0 and 1, with 0 indicating no correlation. Cramér's V is computed as:

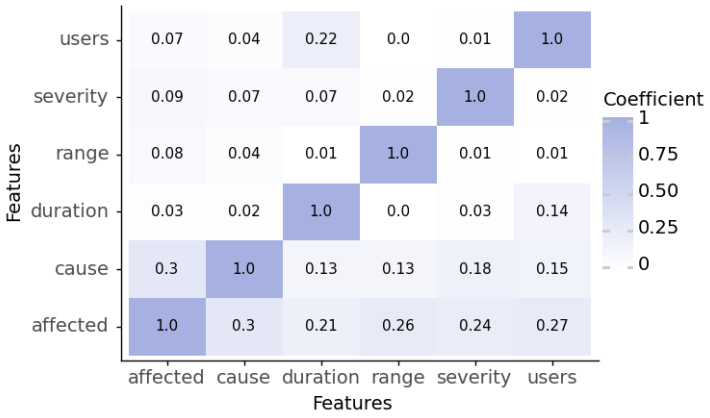

Figure 10: Uncertainty coefficient matrix (Theil's U)

$$V = \sqrt{\frac{\chi^2}{n \times \min(k-1, r-1)}}$$

where r is the number of rows and k the number of columns in the contingency table, n is the number of observations, and χ^2 is derived from Pearson's chi-squared test [14].

Applying the measure to the dataset, we obtain a correlation matrix shown in Figure 9. We observe that there is no strong correlation between any of the features. Relative to the other values in the matrix, the component affected in the outage seems to be moderately correlated with the range of the outage (0.5), and with the users affected in the outage (0.49). There may also be a slight correlation between the cause of the outage and the range of the outage (0.38), as well as with the users affected in the outage (0.31). However, since the values are low, these results are inconclusive.

Association analysis with the uncertainty coefficient (Theil's U) We calculate uncertainty coefficients, also known as Theil's U, for the features of the data; this is shown in a matrix in Figure 10. The uncertainty coefficient of y with respect to x, written as $U(y|x)$, shows how much information x provides about y, and is a value between 0 and 1. A value of 0 indicates that x gives no information about y, and a value of 1 that knowledge of x completely predicts y. Computing Theil's U clarifies the associations seen from Cramér's V, as Cramér's V is symmetric, but this is not necessarily true for the actual correlations. The uncertainty coefficient provides more information about the true relations between the different features [15].

$U(y|x)$ is computed as:

$$U(y|x) = \frac{H(y) - H(y|x)}{H(y)}$$

where H is entropy of a single distribution, and $H(y|x)$ is the entropy of y conditional on x [16].

The results in Figure 10 show no clear association, as the coefficients are low. However, the users affected in the outage could determine the affected component (0.34) and range of the outage (0.3). This seems to support the findings from Figure 9. Similarly, the affected component could determine the cause of the outage (0.28).

5 Threats to Validity

Having presented the results and analysis, this section discusses the main limitations of the study.

1. Manual classification of data points. Due to the lack of structure or standard format among the failure descriptions, all events had to be classified manually across different categories (step 5 in Figure 1). Though the classification was conducted diligently and checked multiple times, such a process is vulnerable to human error. Furthermore, by relying on our interpretation of the descriptions for classification, we introduce an element of subjectivity in the dataset. It is possible that some events are misclassified, which would negatively affect the validity of our results. Unfortunately, because of the aforementioned lack of structure in the reports and insufficient tools for analysis of such data, this is currently unavoidable.

2. Lack of available data. In our analysis, we use a dataset with a relatively small amount of events. To draw more significant and valid conclusions, a much larger dataset would be needed. As no such public repository of data is currently available, we do not have a way to obtain these data. Moreover, for some analyses, we exclude a few data points due to insufficient information. Such selective cleaning may lead to unintended consequences or biases in our results.

3. Heuristic processing errors. For AWS and Azure failures, we extract the event start and end time from textual descriptions. We use heuristic methods based on sentence structures. Despite taking the utmost care, there is a non-zero possibility that we might have missed or wrongly attributed certain failures.

6 Related work

7 Conclusion

References

[1] M. Armbrust, A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "A view of cloud computing," *Commun. ACM*, vol. 53, pp. 50–58, Apr. 2010. DOI: 10.1145/1721654.1721672.

[2] J. Dean, "The rise of cloud computing systems," in *SOSP History Day 2015*, ser. SOSP '15, Monterey, California: Association for Computing Machinery, 2015, ISBN: 9781450340175. DOI: 10.1145/2830903.2830913. [Online]. Available: https://doi.org/10.1145/2830903.2830913.

[3] M. Steen and A. S. Tanenbaum, "A brief introduction to distributed systems," *Computing*, vol. 98, no. 10, pp. 967–1009, Oct. 2016, ISSN: 0010-485X. DOI: 10.1007/s00607-016-0508-7. [Online]. Available: https://doi.org/10.1007/s00607-016-0508-7.

[4] KATC News, *Lafayette 911 moving to cloud-based dispatch system*, https://katc.com/news/around-acadiana/st-martin-parish/2019/03/19/lafayette-911-moving-to-cloud-based-dispatch-system/, Accessed: 2020-06-01, Mar. 2019.

[5] A. Mazmanian and F. Konkel, *Cloud failure temporarily crashes healthcare.gov*, https://fcw.com/articles/2013/10/28/cloud-failure-crashes-healthcare-gov.aspx, Accessed: 2020-06-01, Oct. 2013.

[6] Genaro Network, *Tencent was claimed ten million for data loss due to cloud hard drive glitch*, https://medium.com/genaro-network/tencent-was-claimed-ten-million-for-data-loss-due-to-cloud-hard-drive-glitch-344a26449fe2, Accessed: 2020-06-01, Aug. 2018.

[7] AWS, *AWS status JSON feed*, Accessed: 2020-05-18. [Online]. Available: http://status.aws.amazon.com/data.json.

[8] Google Cloud Platform, *Google Cloud incidents JSON feed*, Accessed: 2020-05-18. [Online]. Available: https://status.cloud.google.com/incidents.json.

[9] Microsoft Azure, *Azure status history*, Accessed: 2020-05-18. [Online]. Available: https://status.azure.com/en-us/status/history/.

[10] H. S. Gunawi, M. Hao, R. O. Suminto, A. Laksono, A. D. Satria, J. Adityatama, and K. J. Eliazar, "Why does the cloud stop computing?: Lessons from hundreds of service outages," in *Proceedings of the Seventh ACM Symposium on Cloud Computing - SoCC '16*, Santa Clara, CA, USA: ACM Press, 2016, pp. 1–16, ISBN: 978-1-4503-4525-5. DOI: 10.1145/2987550.2987583. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2987550.2987583 (visited on 03/02/2020).

[11] H. S. Gunawi, V. Martin, A. D. Satria, M. Hao, T. Leesatapornwongsa, T. Patana-anake, T. Do, J. Adityatama, K. J. Eliazar, A. Laksono, and J. F. Lukman, "What bugs live in the cloud?: A study of 3000+ issues in cloud systems," in *Proceedings of the ACM Symposium on Cloud Computing - SOCC '14*, Seattle, WA, USA: ACM Press, 2014, pp. 1–14, ISBN: 978-1-4503-3252-1. DOI: 10.1145/2670979.2670986. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2670979.2670986 (visited on 03/02/2020).

[12] J. Langford, L. Li, R. P. McAfee, and K. Papineni, "Cloud control: Voluntary admission control for intranet traffic management," *Information Systems and e-Business Management*, vol. 10, pp. 295–308, 2012.

[13] *Choose an AWS region*, https://docs.aws.amazon.com/emr/latest/ManagementGuide/emr-plan-region.html, 2020.

[14] E. W. Holmes, "Handbook of Parametric and Nonparametric Statistical Procedures," *Clinical Chemistry*, vol. 44, no. 11, pp. 2384–2384, Nov. 1998, ISSN: 0009-9147. DOI: 10.1093/clinchem/44.11.2384. eprint: https://academic.oup.com/clinchem/article-pdf/44/11/2384/32727301/clinchem2384.pdf.

[15] S. Zychlinski, *The search for categorical correlation*, https://towardsdatascience.com/the-search-for-categorical-correlation-a1cf7f1888c9, Feb. 2018.

[16] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery, *Numerical Recipes: The Art of Scientific Computing*, 3rd ed. Cambridge University Press, 2007, ISBN: 9780521880688.

A Problem statement

B Self-reflection