CE 528 Cloud Computing

Lecture 16: Performance Profiling
Spring 2026

Prof. Yigong Hu

Slides courtesy of Chang Lou and Robert Morrios

Logistics

Final Presentation is on Apr 27th and Apr 29th

Criteria	Percentage	Expectation
Completeness	50	Have you **built** a tool that can address the problem you proposed in Demo 1?
Code Quality	10	Working code without bugs
Design Doc + Video	20	TA can independently run your code following the doc and demo video
Slides	10	Clearly communicates the improvement you made

Logistics

Final Presentation: 30% of the total grade

- 15% presentation
- 85% project metrics

Two presentation formats:

- In-class presentation: at most three students
- Presentation video (15–20 minutes)
- Each group member must present in at least one format
- If you choose both, we will use the higher score as your presentation score

Logistics

We will send out a link for peer evaluation early next week

- Please submit the evaluation before May 1st
- The evaluation is anonymous
- Any form of collusion in the evaluation will be considered cheating
- We will release scores for quiz, projects, presentations and bonus points in gradescope
- If you believe the final score is incorrect, you may submit a regrade request to review your evaluation score

Why does the Cloud stop computing?

Lessons from hundreds of service outages

Haryadi S. Gunawi, Mingzhe Hao, Riza O. Suminto, Agung Laksono,
Anang D. Satria, Jeffrey Adityatama, and Kurnia J. Eliazar

Availability is Important

Southwest Airlines computer outage grounds fleet nationwide

A Southwest Airlines jet comes in to land at Lindbergh Field in San Diego, California February 25, 2015.

JACK STEWART TRANSPORTATION 08.08.16 7:40 PM

HOW A COMPUTER OUTAGE CAN TAKE DOWN A WHOLE AIRLINE

Passengers await updates from Delta Airlines at Pearson International airport in Toronto, Canada on August 8, 2016. GIORDANO CIAMPINI/ANADOLU AGENCY/GETTY IMAGES

Netflix goes down. Twitter blows up

by Jackie Wattles @jackiewattles

October 2, 2016: 8:23 AM ET

Recommend 1.2K

This Paper Want to Answer

1. How many services do not reach 99% (or 99.9%) availability?

2. Do outages happen more in mature or young services?

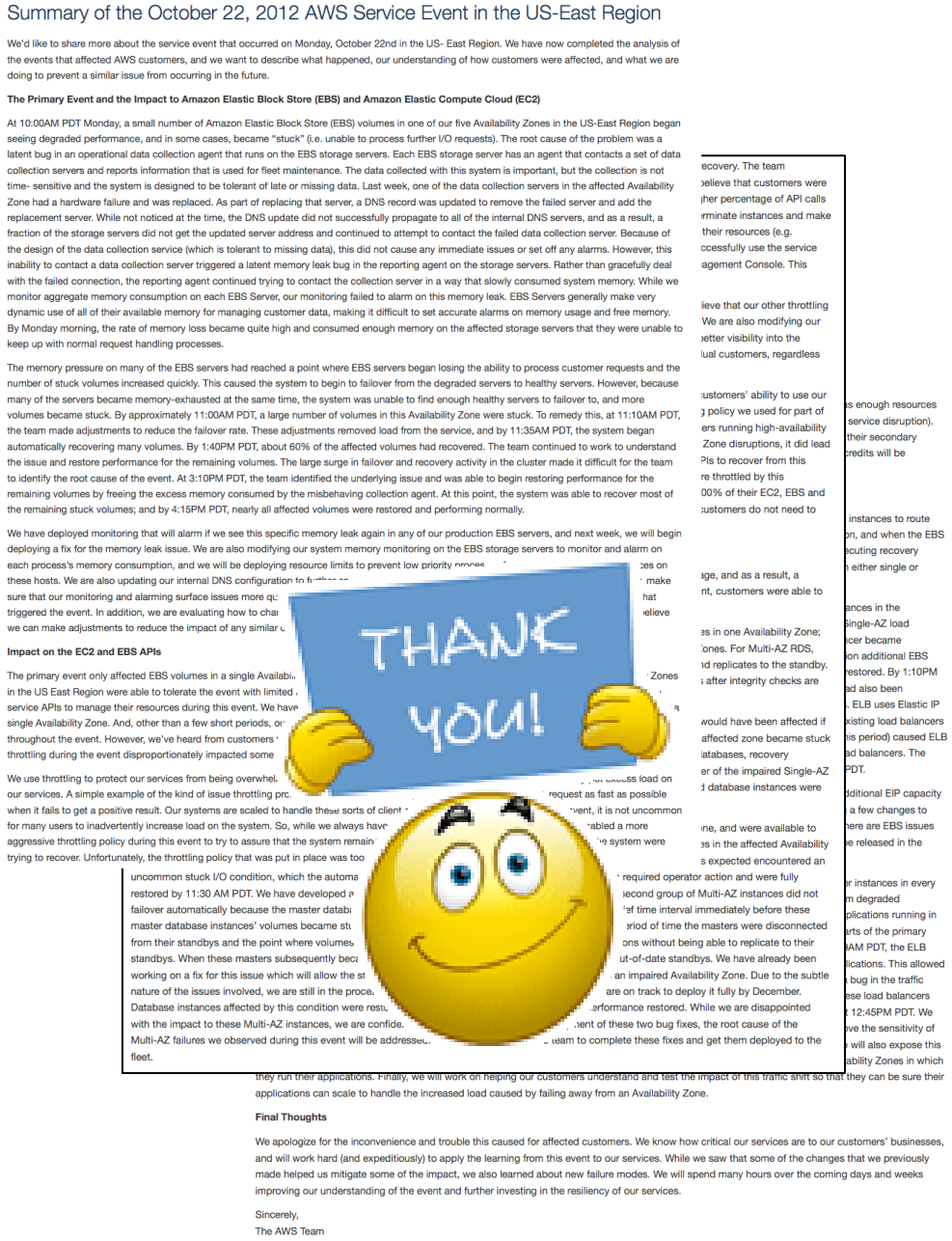
3. What are the common root causes that plague a wide range of service deployments?

4. What are the common lessons that can be gained from various outages?

How to Get the Answer

Public Reports:

- Headline news and post-mortem reports
  - Providers' transparency
  - **Untapped** information
- **Pros/cons**
  - + Detailed root causes
  - + Detailed chain of failures
  - + Downtime durations
  - + Zero false positive
  - -- (Very) incomplete
  - -- (High) variance



Summary of the October 22, 2012 AWS Service Event in the US-East Region

We'd like to share more about the service event that occurred on Monday, October 22nd in the US- East Region. We have now completed the analysis of the events that affected AWS customers, and we want to describe what happened, our understanding of how customers were affected, and what we are doing to prevent a similar issue from occurring in the future.

The Primary Event and the Impact to Amazon Elastic Block Store (EBS) and Amazon Elastic Compute Cloud (EC2)

At 10:00AM PDT Monday, a small number of Amazon Elastic Block Store (EBS) volumes in one of our five Availability Zones in the US-East Region began seeing degraded performance, and in some cases, became "stuck" (i.e. unable to process further I/O requests). The root cause of the problem was a latent bug in an operational data collection agent that runs on the EBS storage servers. Each EBS storage server has an agent that contacts a set of data collection servers and reports information that is used for fleet maintenance. The data collected with this system is important, but the collection is not time- sensitive and the system is designed to be tolerant of late or missing data. Last week, one of the data collection servers in the affected Availability Zone had a hardware failure and was replaced. As part of replacing that server, a DNS record was updated to remove the failed server and add the replacement server. While not noticed at the time, the DNS update did not successfully propagate to all of the internal DNS servers, and as a result, a fraction of the storage servers did not get the updated server address and continued to attempt to contact the failed data collection server. Because of the design of the data collection service (which is tolerant to missing data), this did not cause any immediate issues or set off any alarms. However, this inability to contact a data collection server triggered a latent memory leak bug in the reporting agent on the storage servers. Rather than gracefully deal with the failed connection, the reporting agent continued trying to contact the collection server in a way that slowly consumed system memory. While we monitor aggregate memory consumption on each EBS Server, our monitoring failed to alarm on this memory leak. EBS Servers generally make very dynamic use of all of their available memory for managing customer data, making it difficult to set accurate alarms on memory usage and free memory. By Monday morning, the rate of memory loss became quite high and consumed enough memory on the affected storage servers that they were unable to keep up with normal request handling processes.

The memory pressure on many of the EBS servers had reached a point where EBS servers began losing the ability to process customer requests and the number of stuck volumes increased quickly. This caused the system to begin to failover from the degraded servers to healthy servers. However, because many of the servers became memory-exhausted at the same time, the system was unable to find enough healthy servers to failover to, and more volumes became stuck. By approximately 11:00AM PDT, a large number of volumes in this Availability Zone were stuck. To remedy this, at 11:10AM PDT, the team made adjustments to reduce the failover rate. These adjustments removed load from the service, and by 11:35AM PDT, the system began automatically recovering many volumes. By 1:40PM PDT, about 60% of the affected volumes had recovered. The team continued to work to understand the issue and restore performance for the remaining volumes. The large surge in failover and recovery activity in the cluster made it difficult for the team to identify the root cause of the event. At 3:10PM PDT, the team identified the underlying issue and was able to begin restoring performance for the remaining volumes by freeing the excess memory consumed by the misbehaving collection agent. At this point, the system was able to recover most of the remaining stuck volumes; and by 4:15PM PDT, nearly all affected volumes were restored and performing normally.

We have deployed monitoring that will alarm if we see this specific memory leak again in any of our production EBS servers, and next week, we will begin deploying a fix for the memory leak issue. We are also modifying our system memory monitoring on the EBS storage servers to monitor and alarm on each process's memory consumption, and we will be deploying resource limits to prevent low priority proces… …es on these hosts. We are also updating our internal DNS configuration to f… …make sure that our monitoring and alarming surface issues more qu… …hat triggered the event. In addition, we are evaluating how to chan… …elieve we can make adjustments to reduce the impact of any similar …

Impact on the EC2 and EBS APIs

The primary event only affected EBS volumes in a single Availabi… …Zones in the US East Region were able to tolerate the event with limited … service APIs to manage their resources during this event. We have … single Availability Zone. And, other than a few short periods, o… …would have been affected if throughout the event. However, we've heard from customers … …affected zone became stuck throttling during the event disproportionately impacted some … …atabases, recovery

We use throttling to protect our services from being overwhel… …request as fast as possible our services. A simple example of the kind of issue throttling pro… …vent, it is not uncommon when it fails to get a positive result. Our systems are scaled to handle these sorts of client … …abled a more for many users to inadvertently increase load on the system. So, while we always have … …e system were aggressive throttling policy during this event to try to assure that the system remain… trying to recover. Unfortunately, the throttling policy that was put in place was too … …s expected encountered an uncommon stuck I/O condition, which the automa… …required operator action and were fully restored by 11:30 AM PDT. We have developed a … …econd group of Multi-AZ instances did not failover automatically because the master databa… …f time interval immediately before these master database instances' volumes became stu… …riod of time the masters were disconnected from their standbys and the point where volumes … …ons without being able to replicate to their standbys. When these masters subsequently beca… …ut-of-date standbys. We have already been working on a fix for this issue which will allow the st… …an impaired Availability Zone. Due to the subtle nature of the issues involved, we are still in the proce… …are on track to deploy it fully by December. Database instances affected by this condition were resu… …erformance restored. While we are disappointed with the impact to these Multi-AZ instances, we are confide… …ent of these two bug fixes, the root cause of the Multi-AZ failures we observed during this event will be addresse… …eam to complete these fixes and get them deployed to the fleet.

…ecovery. The team …elieve that customers were …her percentage of API calls …rminate instances and make their resources (e.g. …ccessfully use the service …agement Console. This

…ieve that our other throttling We are also modifying our …etter visibility into the …lual customers, regardless

…ustomers' ability to use our …g policy we used for part of …ers running high-availability Zone disruptions, it did lead …PIs to recover from this …re throttled by this …00% of their EC2, EBS and …ustomers do not need to

…ge, and as a result, a …nt, customers were able to

…es in one Availability Zone; …ones. For Multi-AZ RDS, …d replicates to the standby. …after integrity checks are

…s enough resources … service disruption). …their secondary …credits will be

… instances to route …on, and when the EBS …ecuting recovery …n either single or

…ances in the …Single-AZ load …cer became …on additional EBS …restored. By 1:10PM …ad also been …. ELB uses Elastic IP …xisting load balancers …his period) caused ELB …ad balancers. The …PDT.

…dditional EIP capacity … a few changes to …here are EBS issues …e released in the

…ne, and were available to …es in the affected Availability

…r instances in every …m degraded …plications running in …arts of the primary …AM PDT, the ELB …lications. This allowed … bug in the traffic …ese load balancers … 12:45PM PDT. We …ve the sensitivity of … will also expose this …ability Zones in which they run their applications. Finally, we will work on helping our customers understand and test the impact of this traffic shift so that they can be sure their applications can scale to handle the increased load caused by failing away from an Availability Zone.

Final Thoughts

We apologize for the inconvenience and trouble this caused for affected customers. We know how critical our services are to our customers' businesses, and will work hard (and expeditiously) to apply the learning from this event to our services. While we saw that some of the changes that we previously made helped us mitigate some of the impact, we also learned about new failure modes. We will spend many hours over the coming days and weeks improving our understanding of the event and further investing in the resiliency of our services.

Sincerely,
The AWS Team

Cloud Outage Study

32 services

597 outages

- between 2009-2015
- ~70% report downtimes
- ~60% report root causes

Category	Service Names
CH: Chat	Blackberry Messenger, Google Hangouts, Skype, WeChat, WhatsApp
EC: E-Comm.	Amazon.com, Ebay
ML: Email	GMail, Hotmail, Yahoo Mail
GM: Game	PS Network, Xbox Live
PA: PaaS/IaaS	Amazon EBS, EC2, and RDS, Google Ap-pengine, Microsoft Azure, Rackspace
SA: SaaS	Google Docs, Office365, Salesforce
SC: Social	Facebook, Google Plus, Instagram, Twitter
DT: Storage	Apple iCloud, Box, Dropbox, Google Drive, Microsoft SkyDrive
VD: Video	Netflix, Youtube

Disclaimers: As availability is a sensitive matter to service providers, it is important for us to make several disclaimers. First, our study is not meant to discredit any service. Readers should take the high-level lessons but prevent themselves to compare service uptimes (*e.g.*, X is better than Y). For this reason, we anonymize service names (*e.g.*, CH2) when presenting numerical findings. Second, the more popular a service is, the more attention its outages will gather, hence more headlines. In fact, more popular services tend to be more transparent and provide detailed reports that we could learn from. For this reason, again, readers should not use this paper to claim a service is better than others. The pros and cons of our methodology will be presented in Section 7.

Answer for First Question

1.How many services do not reach 99% (or 99.9%) availability?

On average

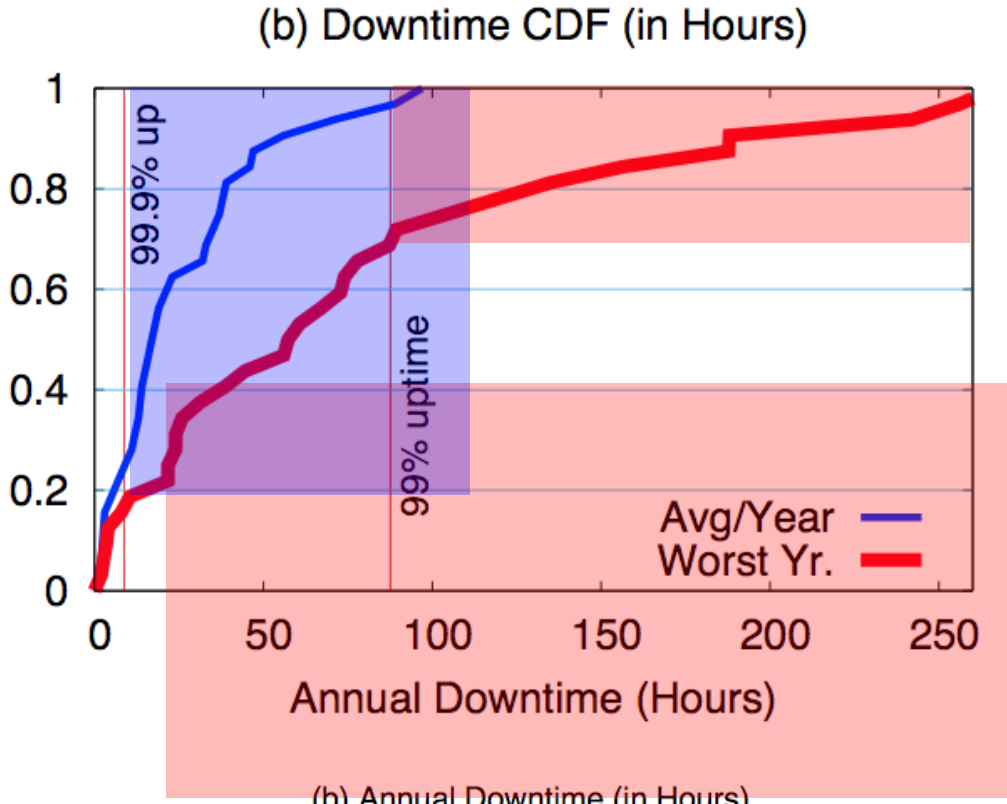
- 6% services do not reach 99% availability (>88 hours)
- 78% not reach 99.9% (>8.8 hours)

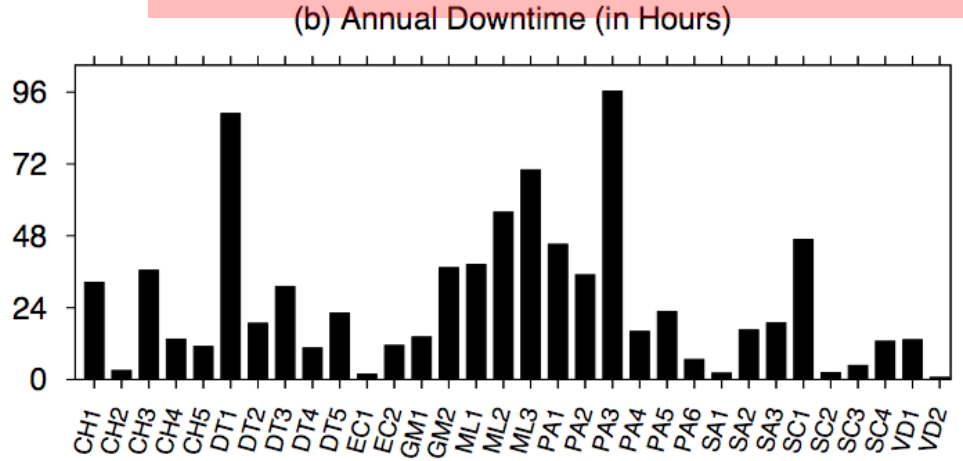
Worst year

- 31% not reach 99%
- 81% ot reach 99.9%

5-nine availability?

- It's just a dream?





Answer for the Second Question

Total outage count from X-year old services in 20YY

	2	3	4	5	6	7	8	9	10	12	15	17	18
’15	1	8	5	5	2	2	26	3	7	8		3	2
’14	4	9	8	14	6		18	24	11	4	1	5	1
’13	7	16	5	8	1	5	25	5	18	5	14	2	4
’12	2	11	5	14	2		23	2	8	4	8	6	1
’11		6	3	11	2	1	16	2	5	3		6	2
’10			2	3	4		28	2	6	6	2	4	3
’09				4	1		60	1	5	4	1	3	2

X: Service Age

Total downtime from X-year old services in 20YY

	2	3	4	5	6	7	8	9	10	12	15	17	18
’15	1	50	12	12	18		20	3	6			1	6
’14	7	37	32	51	67		80	69	14	87	6	267	9
’13	21	270	20	34	0	89	39	1	40	32	72	121	188
’12	0	51	80	46	10		42	2	5	48	25	36	1
’11		25	74	216	135	3	143	1	203	24		55	3
’10			3	57	14		74	2	13	12	1	9	122
’09				156	6		32	1	247	32	0	20	9

X: Service Age

Root Causes

What are the common root causes that plague a wide range of service deployments?

Root cause	Cnt	%
UNKNOWN	355	-
UPGRADE	54	16
NETWORK	52	15
BUGS	51	15
CONFIG	34	10
LOAD	31	9
CROSS	28	8
POWER	21	6
SECURITY	17	5
HUMAN	14	4
STORAGE	13	4
SERVER	11	3
NATDIS	9	3
HARDWARE	5	1

Interesting Root Causes

Upgrade

- Involves **multi-layers**
  - “a code push behaved differently in widespread use than it had during testing”
- To understand/reproduce, need **full ecosystem**

Root cause	Cnt	%
UNKNOWN	355	-
UPGRADE	54	16
NETWORK	52	15
BUGS	51	15
CONFIG	34	10
LOAD	31	9
CROSS	28	8
POWER	21	6
SECURITY	17	5
HUMAN	14	4
STORAGE	13	4
SERVER	11	3
NATDIS	9	3
HARDWARE	5	1

Cloud Storage

Human mistakes

- Rare now (vs. 10 years ago)

Config/Upgrade software bugs

- Bugs in automation process
- Similar issues?
  - But root cause origins are different

Root cause	Cnt	%
UNKNOWN	355	-
UPGRADE	54	16
NETWORK	52	15
BUGS	51	15
CONFIG	34	10
LOAD	31	9
CROSS	28	8
POWER	21	6
SECURITY	17	5
HUMAN	14	4
STORAGE	13	4
SERVER	11	3
NATDIS	9	3
HARDWARE	5	1

Config vs. Upgrade Research

Upgrade #1, need more research?

Paper count in last few years →

Challenges:

- Multi-layer
  - Full ecosystem needed
  - Multi-year?
- Reproducible bugs from industry (benchmarks)?

Conference	Config papers	Upgrade papers
ASPLOS	0	1
ATC	6	2
DSN	8	2
EuroSys	3	2
NSDI	3	0
OSDI	4	0
SOSP	3	1
…		
Total	**27**	**8**

Interesting Root Causes

Bugs

- What types of bugs lead to outages? Why are not masked?
- *(pls. see paper)*
- "Cascading" bugs

Root cause	Cnt	%
UNKNOWN	355	-
UPGRADE	54	16
NETWORK	52	15
BUGS	51	15
CONFIG	34	10
LOAD	31	9
CROSS	28	8
POWER	21	6
SECURITY	17	5
HUMAN	14	4
STORAGE	13	4
SERVER	11	3
NATDIS	9	3
HARDWARE	5	1

Case 1

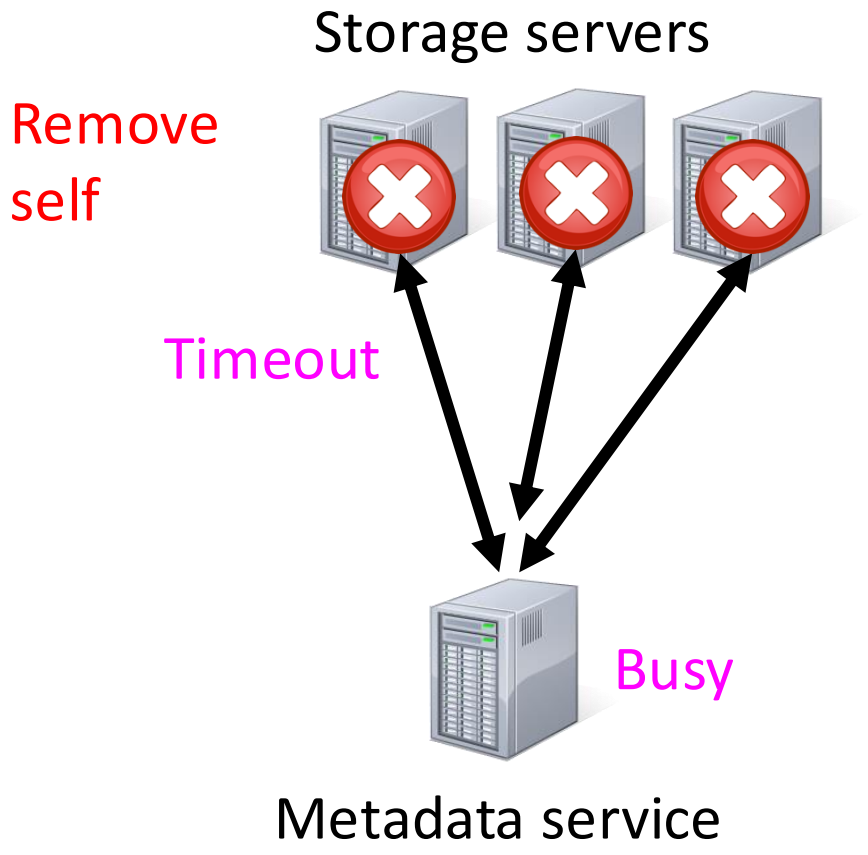
NEWS
9/22/2015
09:31 AM

Amazon Disruption Produces Cloud Outage Spiral

"DynamoDB **Storage servers** query the **metadata service** for their membership"

"But, on Sunday morning, the metadata service responses exceeded the retrieval time allowed by storage servers [**busy timeout**]"

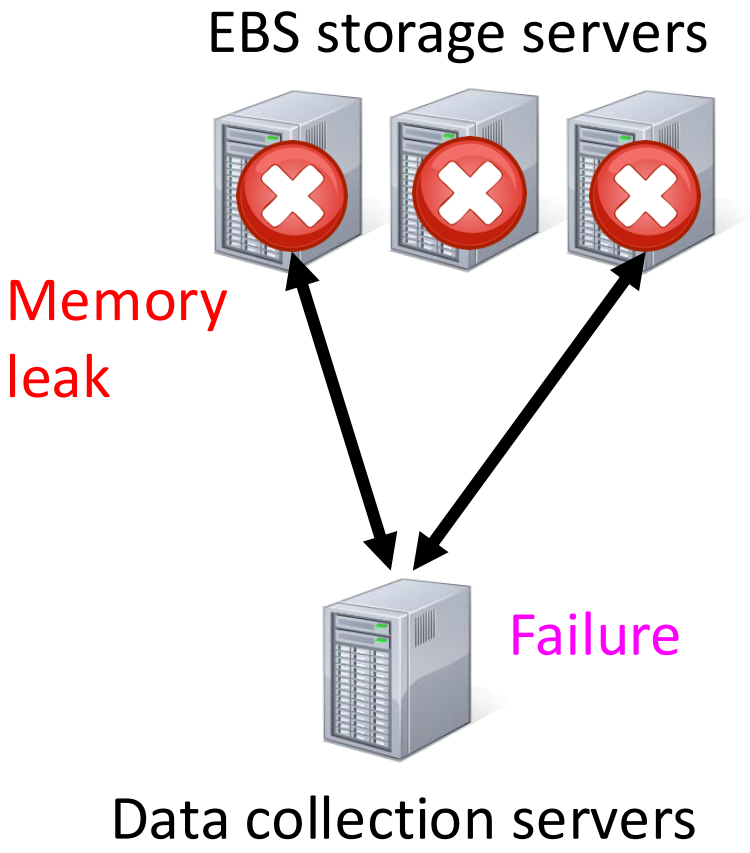
"As a result, the storage servers were unable to obtain their membership data, and **removed** themselves from taking requests"



Case 2

Software Bug, Cascading Failures Caused Amazon Outage

- "Each **EBS storage server** contacts **data collection servers** and reports information that is used for fleet maintenance"
- "data collection servers … had a **failure**"
- "this inability to contact a data collection server triggered **a latent memory leak bug** in the storage servers …
- "EBS servers continued trying in a way that slowly **consumed system memory**"

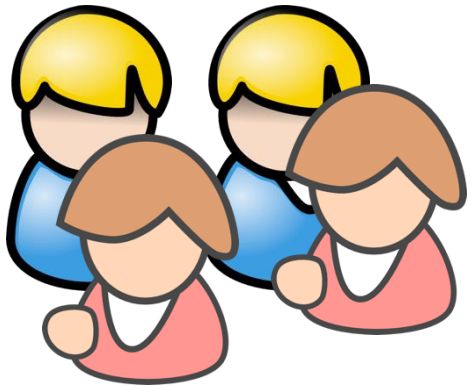


Root cause	#Sv	Cnt	%	Cnt '09-'15
UNKNOWN	29	355	-	M.M.M.M.M.M.M
UPGRADE	18	54	16	7.4.M.5.M.4.7
NETWORK	21	52	15	4.4.6.8.M.8.5
BUGS	18	51	15	M.4.9.8.9.9.2
CONFIG	19	34	10	2.2.7.2.5.M.4
LOAD	18	31	9	2.5.5.5.4.8.2
CROSS	14	28	8	-.2.4.M.5.3.4
POWER	11	21	6	5.4.3.5.3.1.-
SECURITY	9	17	5	7.-.2.1.3.4.-
HUMAN	11	14	4	-.1.4.4.2.1.2
STORAGE	4	13	4	2.-.-.3.5.3.-
SERVER	6	11	3	-.3.-.2.2.4.-
NATDIS	5	9	3	1.1.3.2.1.1.-
HARDWARE	4	5	1	1.-.-.3.1.-.-

Where Is the Single Point of Failure?

Redundancies, redundancies, redundancies!

Yes, we did that



So, why do outages still happen?

Failure Recovery Chain

Failure **Detection**

Failover

Backups

Failure Recovery Chain

Incomplete Failure Detection

Failover that **Fails**

Backups that also **Fail**

Imperfect Failure Recovery Chain

Incomplete error/failure detection

- Undetected (specific type of) memory leaks
- Load spikes of authentication requests
- “an unexpected hardware behavior”

Failure Recovery Chain

Failover/recovery that fails

- Bad PLC fails to activate backup power generators
- Failed network switch failover
- DC failover fails due to *cold cache* problems
- Recovery/re-mirroring storm

Failure Recovery Chain

Multiple failures!

- **Double** failures of power, network, storage or server components
- **Diverse** failures: network+server; storage+fibre cut

Cascading bugs …

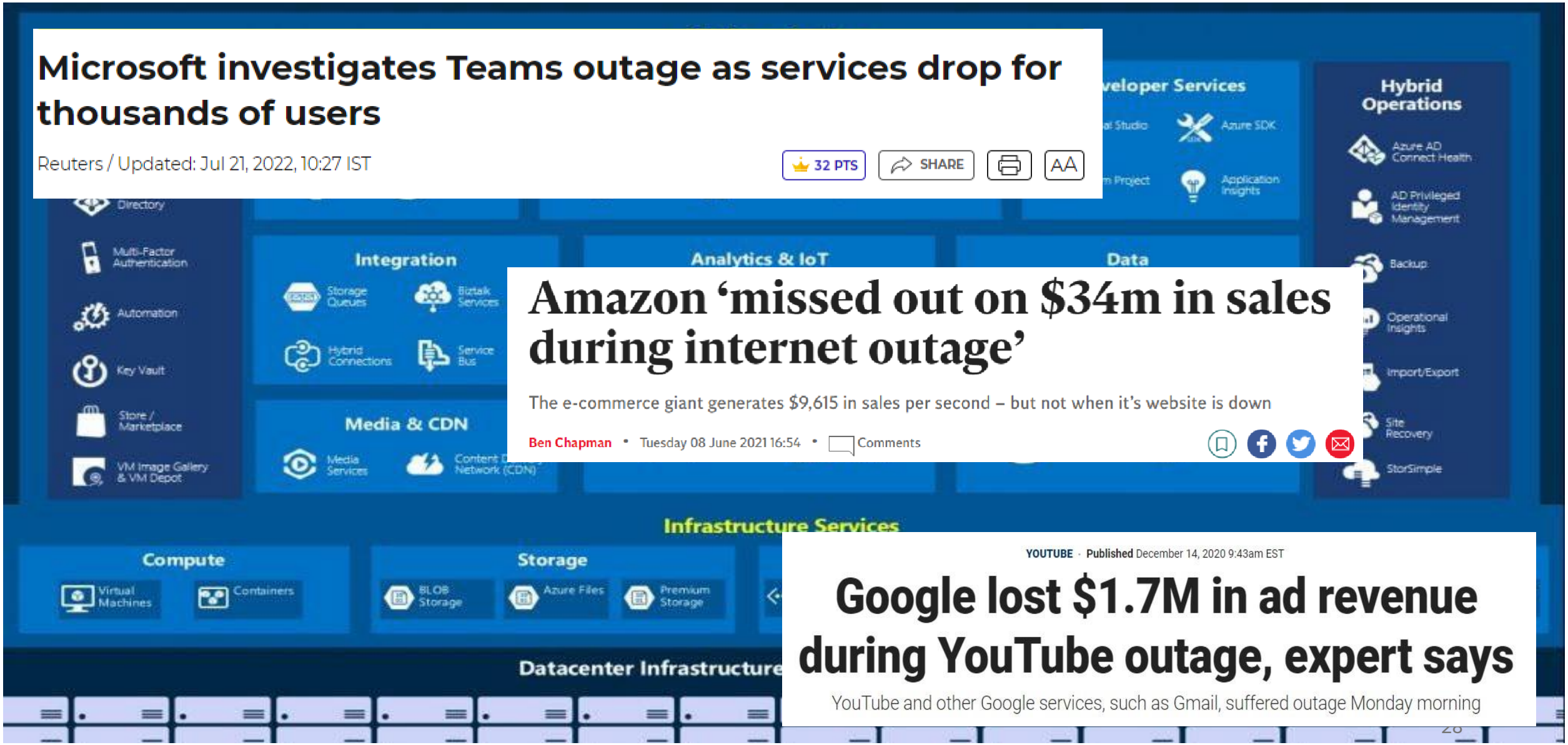
- … that caused many/all redundancies to fail

How to Fight Production Incidents? An Empirical Study on a Large-scale Cloud Service

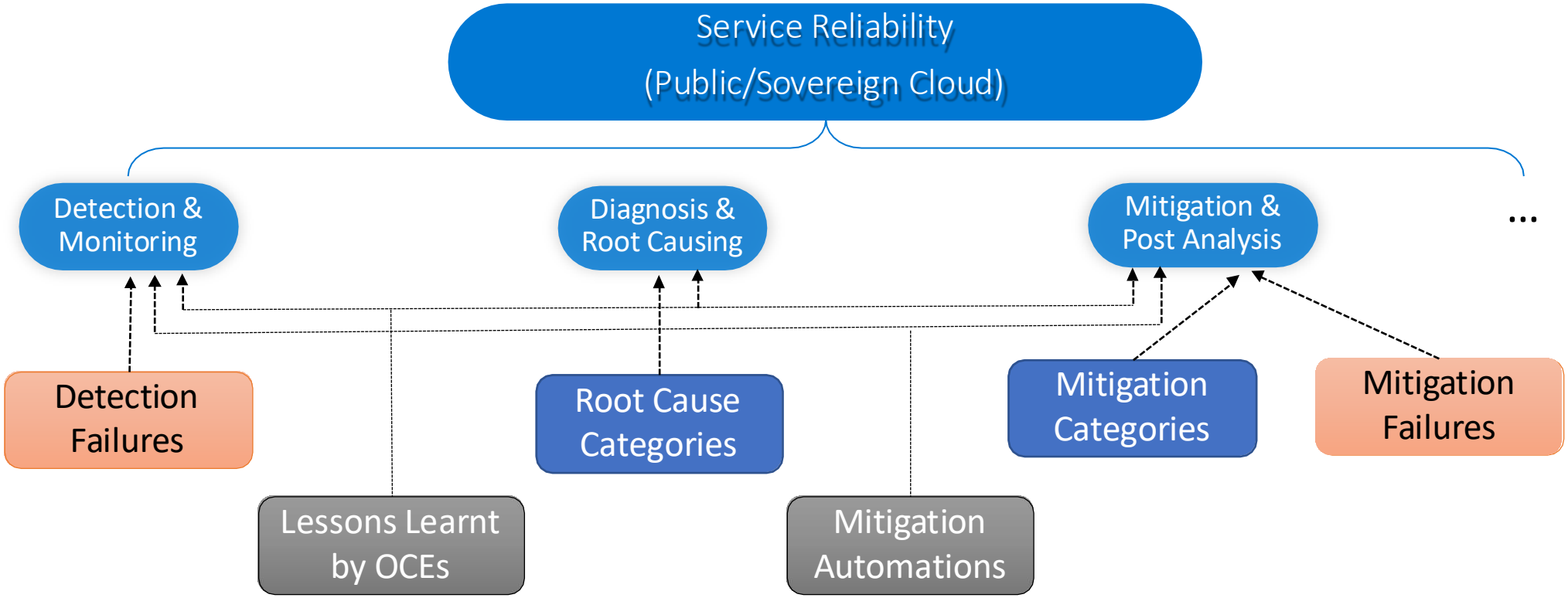
Supriyo Ghosh, Manish Shetty, Chetan Bansal, Suman Nath

Microsoft

Cloud Service Incidents Are Inevitable and Costly



Questions



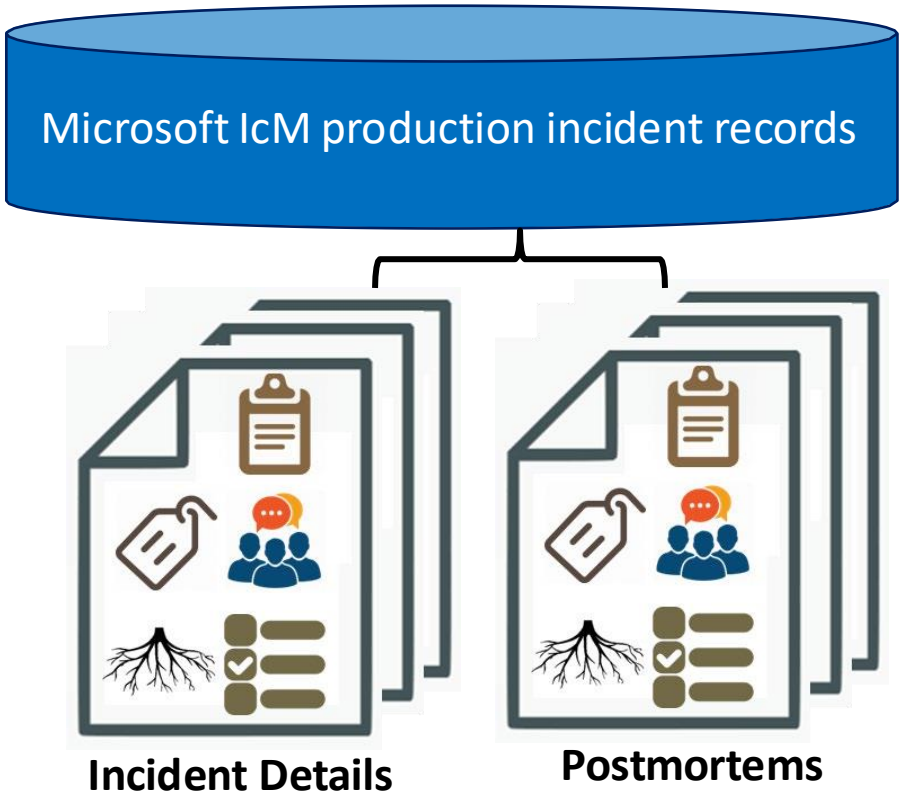
Questions We Aim to Address

1. *Why the incidents occurred and how they were resolved?*
2. *What the gaps were in current processes which caused delayed response?*
3. *What automation could help make the services resilient?*

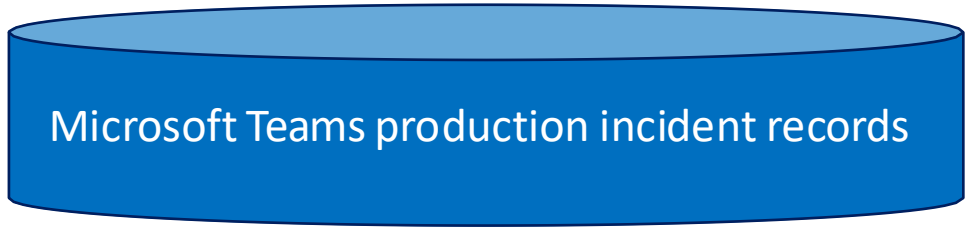
Methodology

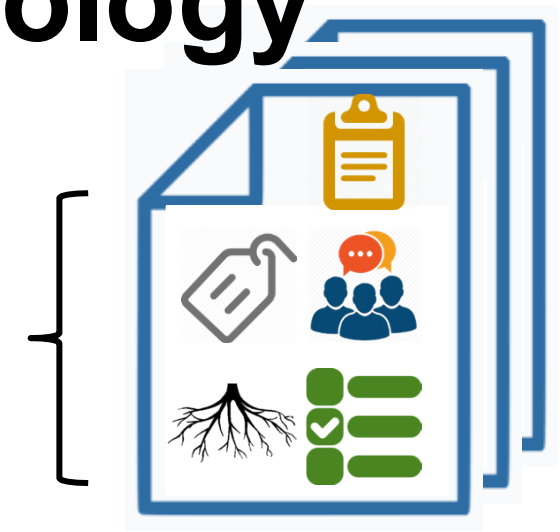
ICM

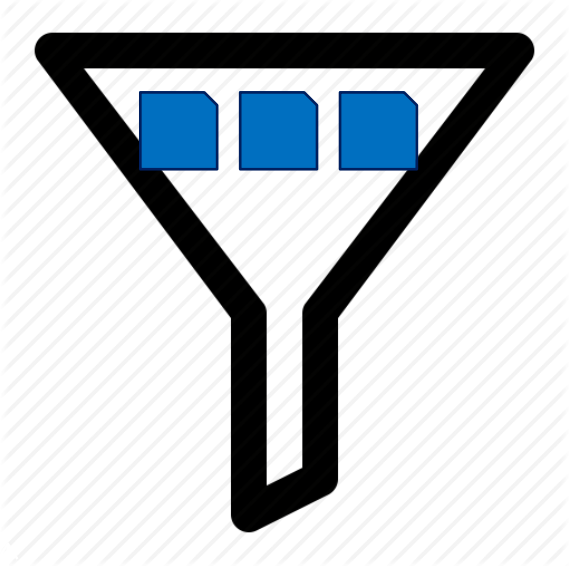
- Incident Management tool (IcM) has plethora of rich information for recent high severity production incidents.
- Post-mortem reports contain useful structured and unstructured information regarding root cause and mitigation.

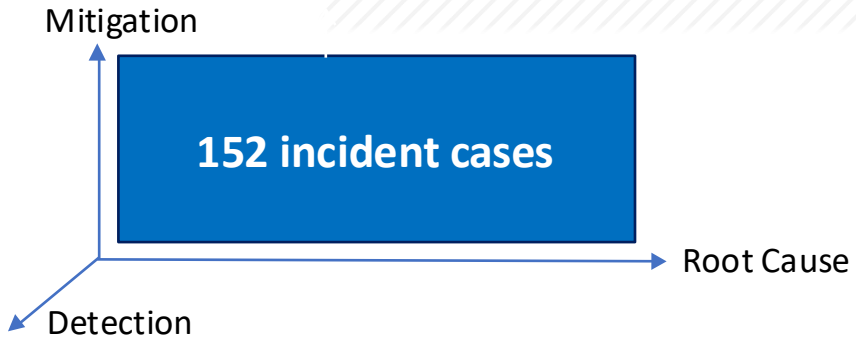


Methodology









- **Incidents** from one year period (05/15/2021 to 05/15/2022)
- Microsoft Teams service
- a feature-blocker or outage incident (high severity)
- incident has been resolved/mitigated
- contains detailed root cause information
- postmortem contains mitigation and discussion

Categorization Strategy

- Dataset split: taxonomy (60 incidents); validation (30 incidents); test set (62 incidents)
- For each of the 6 dimensions
  - Populate summarized text from incident summary and post-mortem reports.
  -  Individually labels categories on taxonomy set
  -  Identify common taxonomy via discussion
  -  Individually labels categories on validation set.
  -  Finalize taxonomy set via discussion

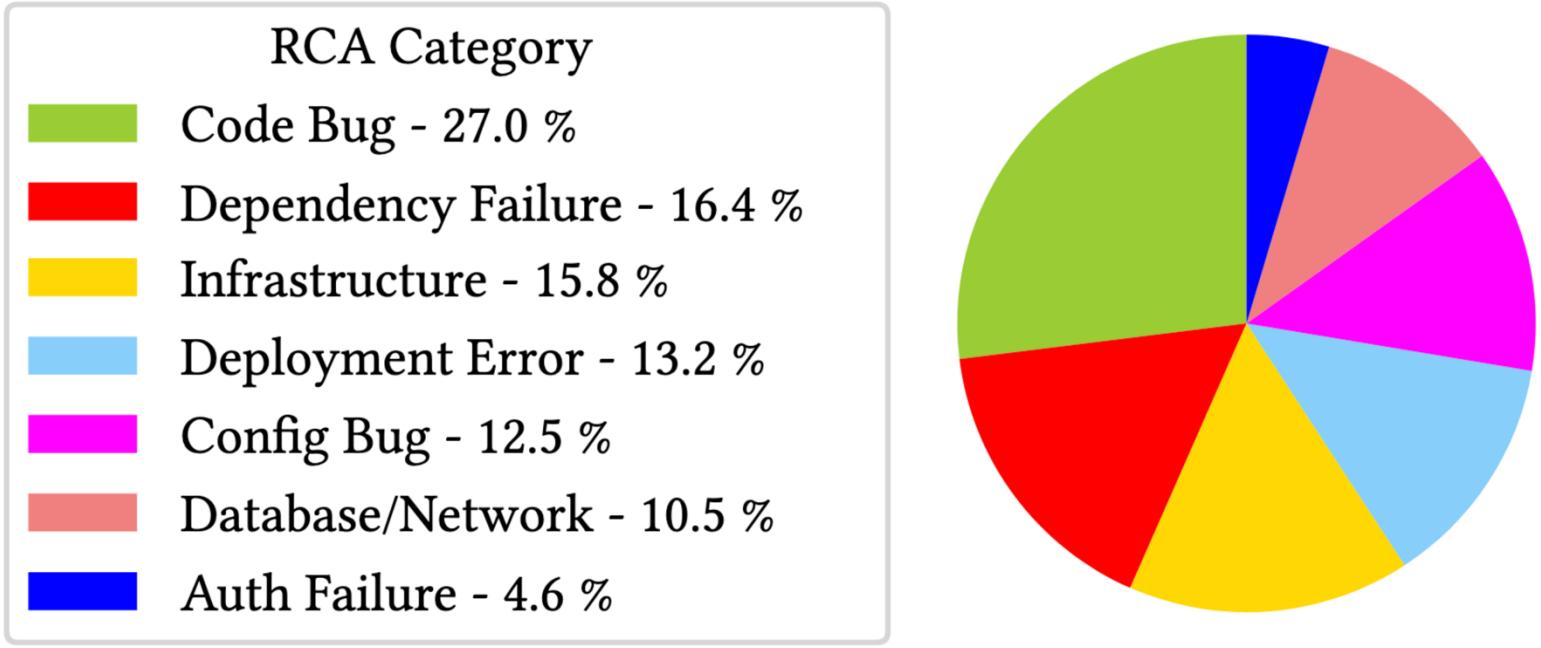
- Root causes
- Mitigation steps
- Detection failures
- Mitigation failures
- Lessons learnt by OCEs
- Automation opportunities

Categorization Strategy

- Dataset split: taxonomy (60 incidents); validation (30 incidents); test set (62 incidents)
- For each of the 6 dimensions
  - Populate summarized text from incident summary and post-mortem reports.
  -  Individually labels categories on taxonomy set
  -  Identify common taxonomy via discussion
  -  Individually labels categories on validation set.
  -  Finalize taxonomy set via discussion
  -  Individually labels categories on test data set
  -  Use **Kohen's kappa** to compute inter-annotator agreement scores (1 is optimal).

- Root causes **(0.94)**
- Mitigation steps **(0.95)**
- Detection failures **(0.88)**
- Mitigation failures **(0.94)**
- Lessons learnt by OCEs **(0.94)**
- Automation opportunities **(0.98)**

Insight from Root Causes



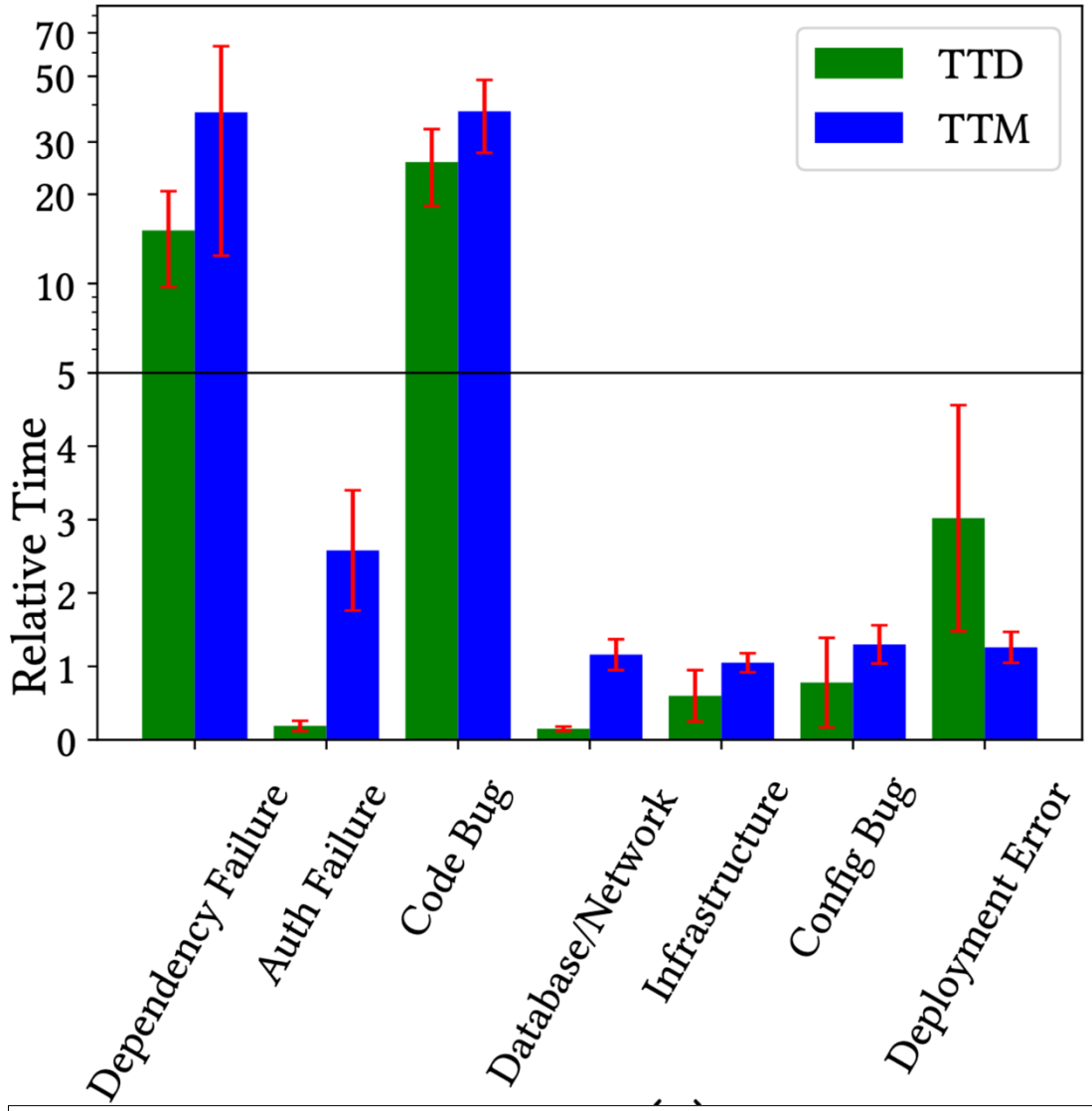
Observation: Majority of incidents (60%) were caused due to non-code/non-config related issues in infrastructure, deployment, and service dependencies.

Implication: Effective techniques need to developed for reliable infra management and safe deployment.

TTD and TTM for Different Root Causes

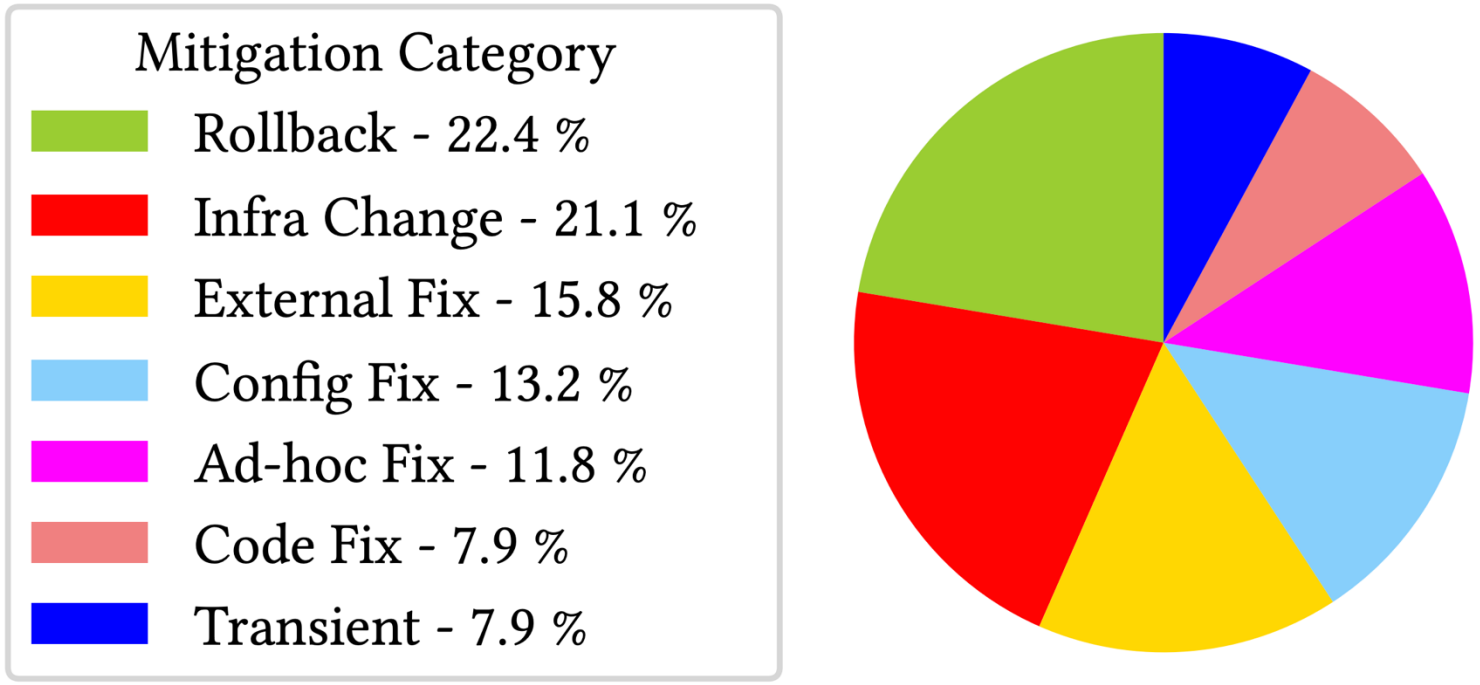
Observation: The time to detect and mitigate code bugs and dependency failures is significantly higher than other root causes.

Implication: We need better observability tool across partner services for better coverages.



Y-axis shows the normalized time, with the median of time to detect or mitigate of all incidents as 1.

Insight from Mitigation Steps



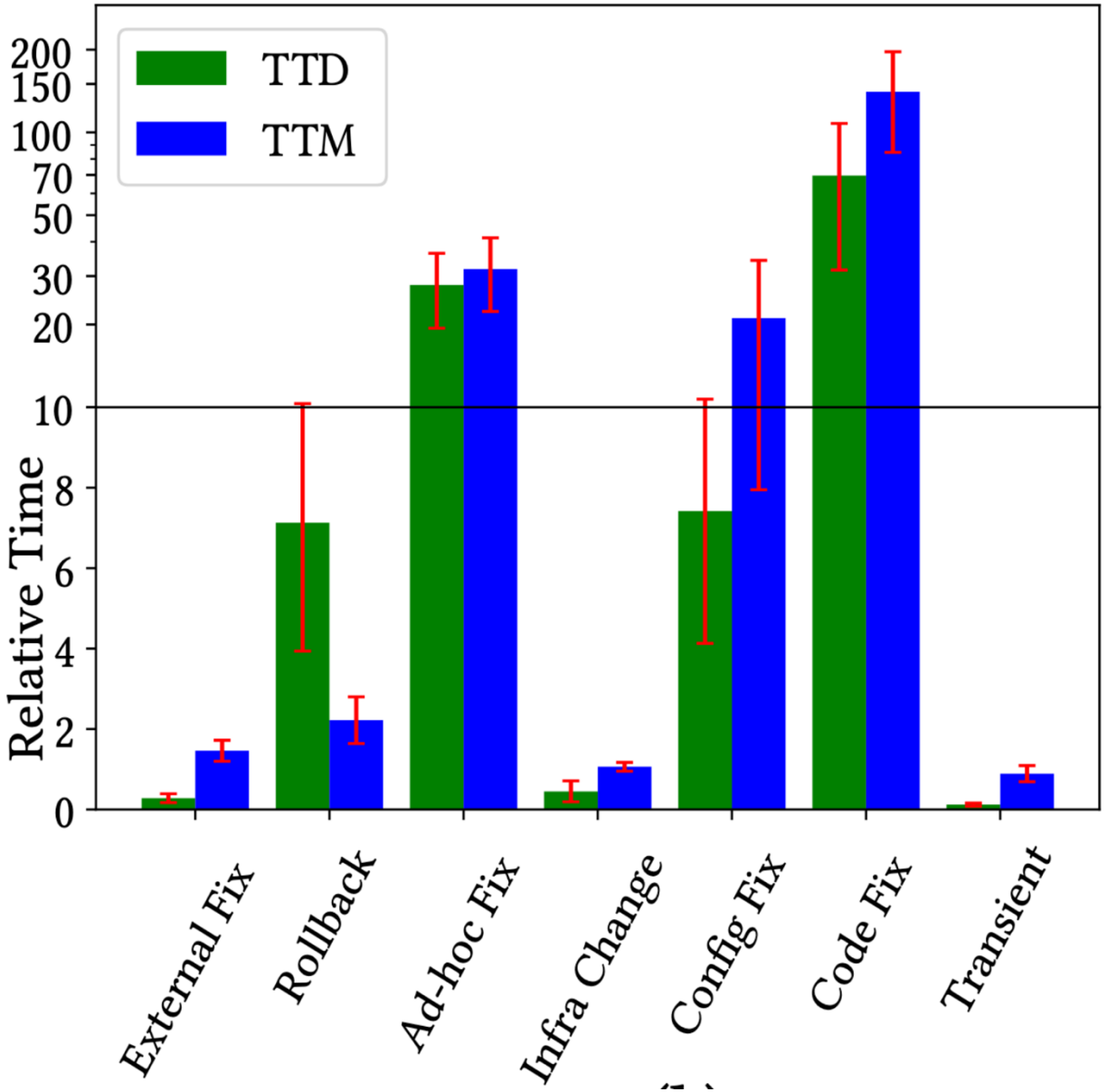
Observation: Among the 40% incidents that were caused by code/configuration bugs, nearly 80% of incidents were mitigated *without* a code or configuration fix.

Implication: We need more effective automation such as auto scaling and auto traffic failover that can mitigate 40% of code/config bugs.

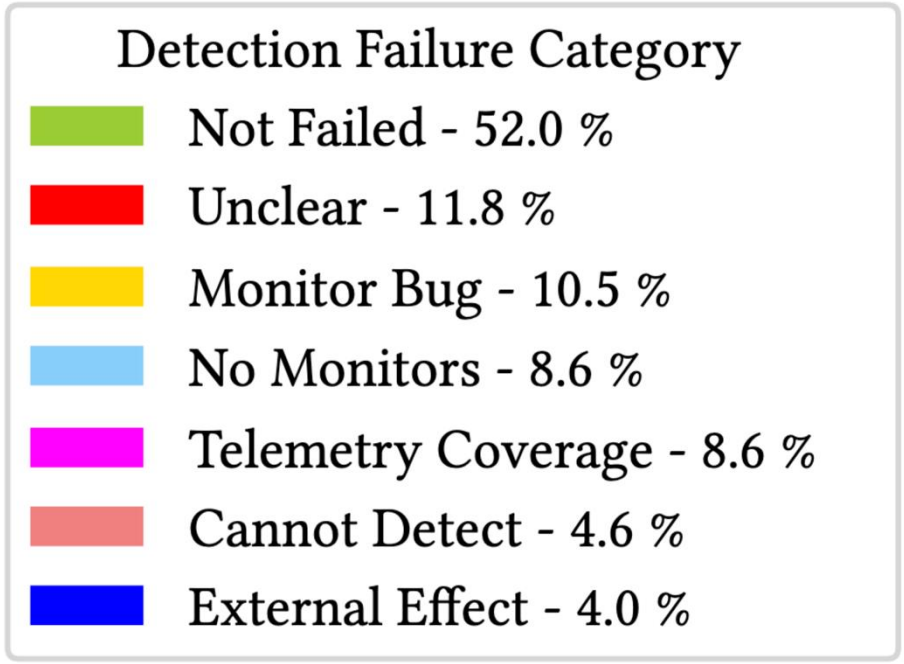
TTD and TTM for Different Mitigation Steps

Observation: 30% of the mitigation delay is caused due to manual mitigation steps

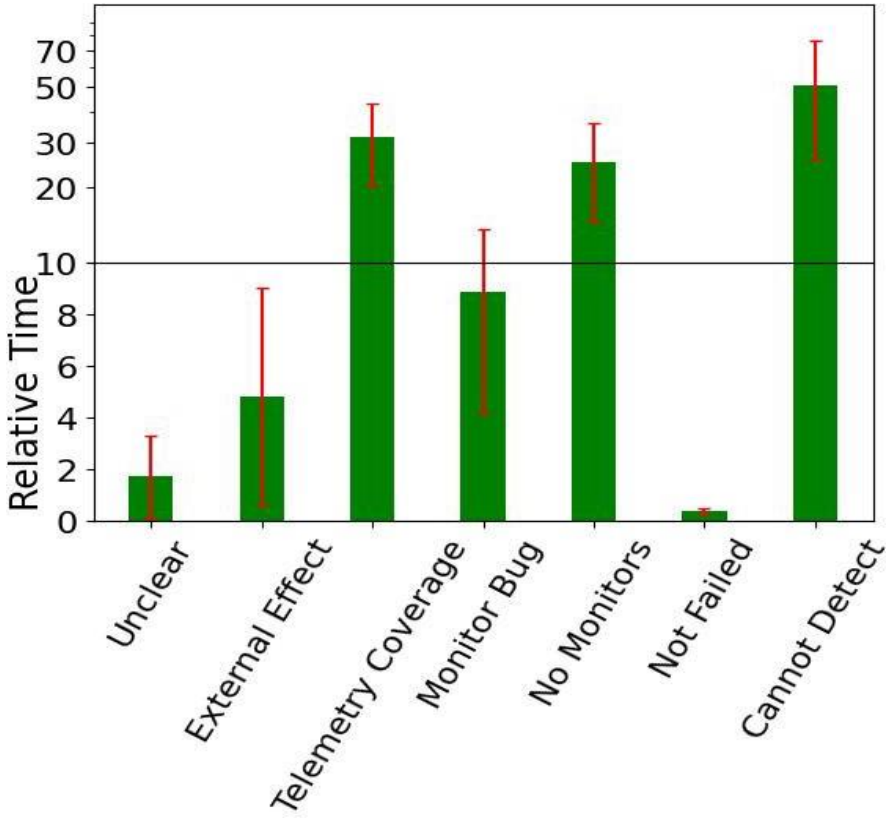
Implication: We need automation tools to reduce human involvement.



Insight from Detection Failures



TTD for different detection failures



Observation: ≈17% of incidents either **lacked monitors or telemetry coverage**. 10% incidents were not detected **due to bugs**, e.g., high threshold, buggy feature, wrong configuration, etc.

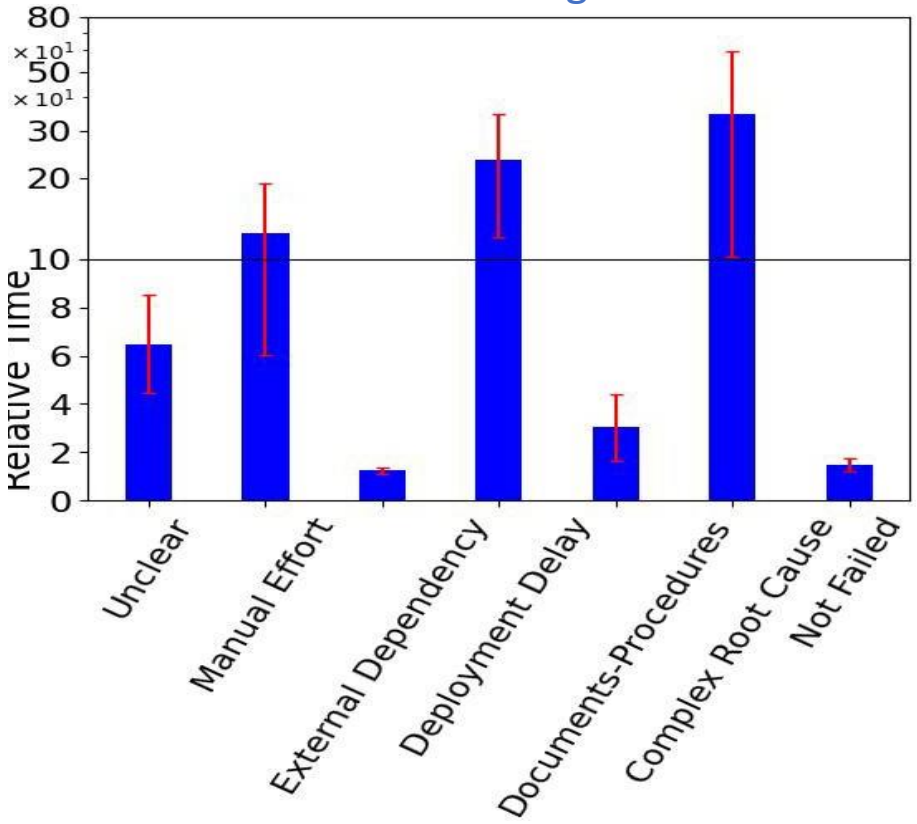
Implication: New watchdogs need be setup with dynamic thresholding mechanism.

Insights from Mitigation Failures

Detection Failure Category

- Not Failed - 52.0 %
- Unclear - 11.8 %
- Monitor Bug - 10.5 %
- No Monitors - 8.6 %
- Telemetry Coverage - 8.6 %
- Cannot Detect - 4.6 %
- External Effect - 4.0 %

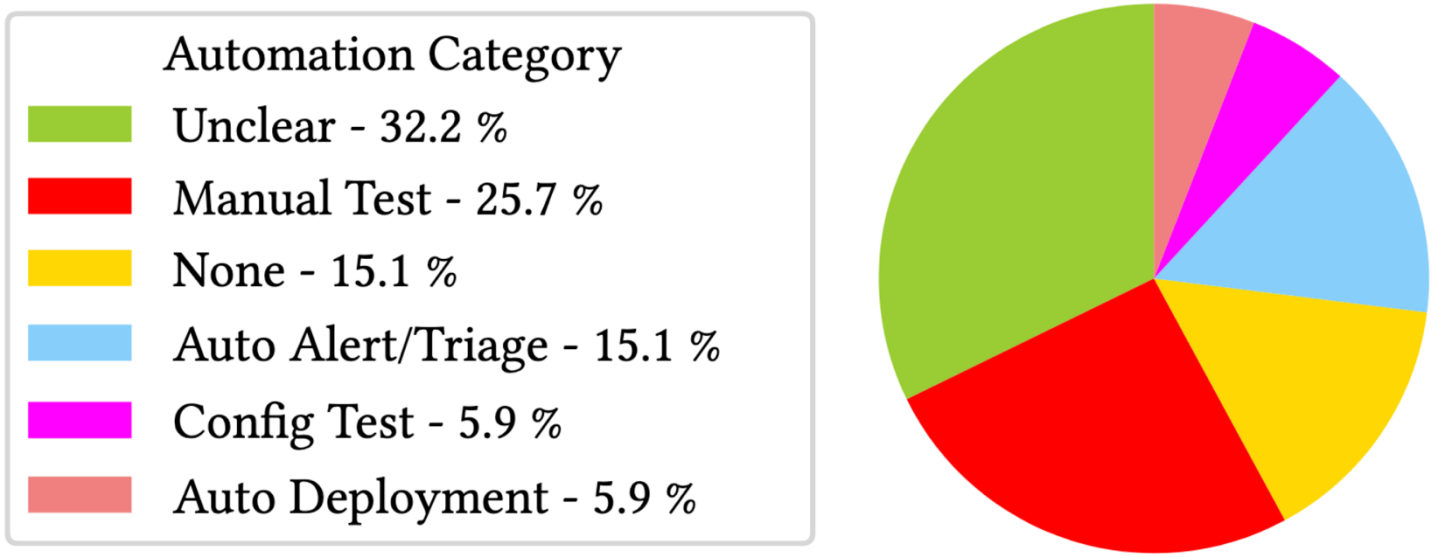
TTM for different mitigation failures



Observation: While 7% mitigation delays are due to complex root causes, 27% of incidents had mitigation delays due to **manual efforts, external dependency and deployment issues.**

Implication: Reducing human intervention through automation can significantly reduce mitigation delay.

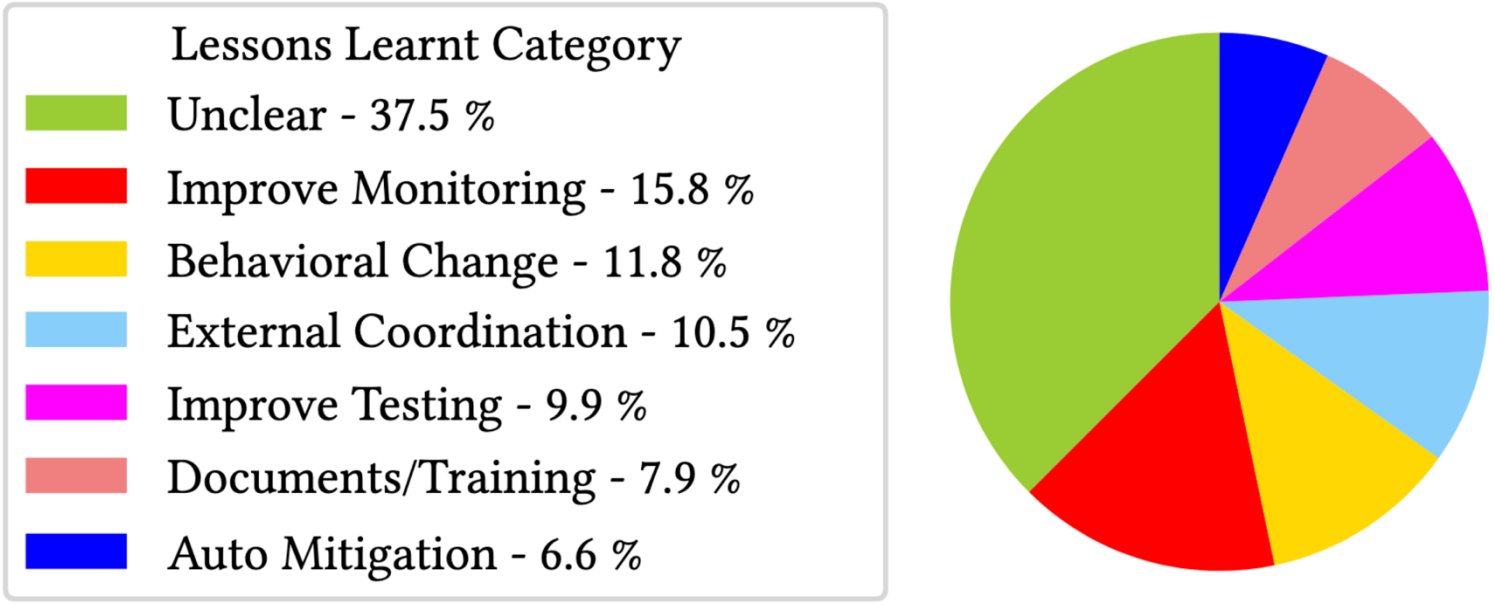
Insights from Automation Suggestions By OCEs



Observation: Improving testing was a popular choice for automation opportunities, over monitoring.

Implication: We need to reduce incidents by identifying issues before they reach production services through automated testing.

Insights from Lessons Learnt By OCEs



Observation: While improving monitoring/testing accounts for majority of the lessons learnt, a significant ≈20% feedback indicated problems with existing documentations.

Implication: We need better documentations, training, and practices for better incident management and service resiliency.