Automatic Root Cause Analysis via Large Language Models for Cloud Incidents

Yinfang Chen, Huaibing Xie, Minghua Ma, Yu Kang, Xin Gao, Liu Shi, Yunjie Cao, Xuedong Gao, Hao Fan, Ming Wen, Jun Zeng, Supriyo Ghosh, Xuchao Zhang, Chaoyun Zhang, Qingwei Lin, Saravan Rajmohan, Dongmei Zhang, and Tianyin Xu

Microsoft

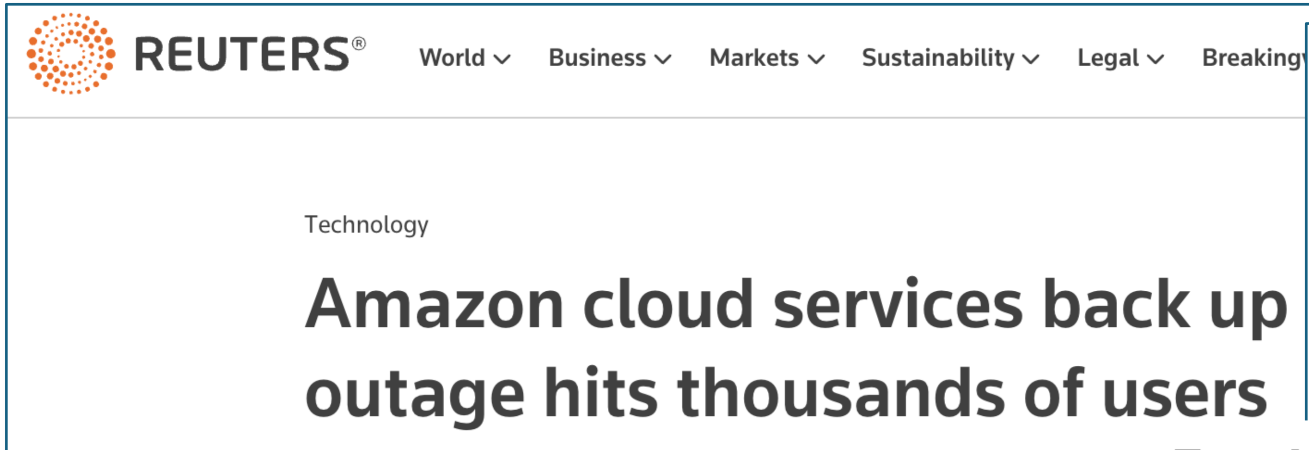
UNIVERSITY OF ILLINOIS URBANA-CHAMPAIGN

北京大學 PEKING UNIVERSITY

華中科技大學 HUAZHONG UNIVERSITY OF SCIENCE AND TECHNOLOGY

NUS National University of Singapore

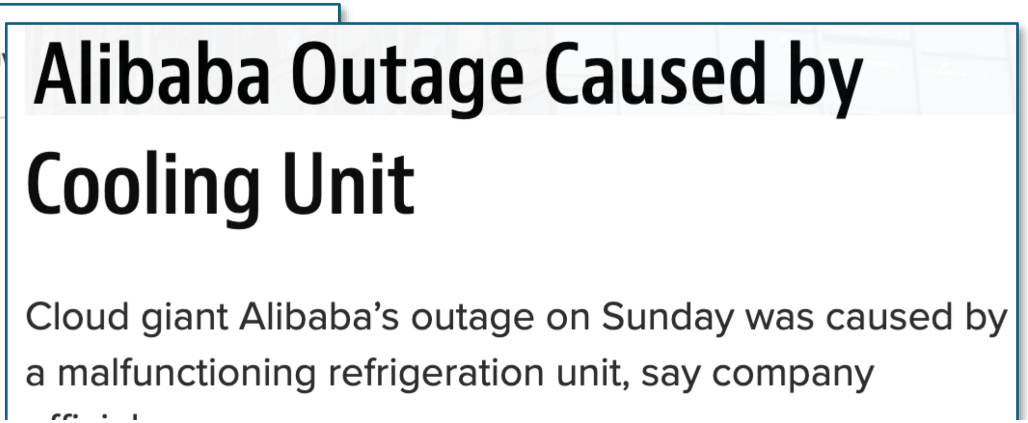
Cloud Incidents are on the Rise



REUTERS® World ⌄ Business ⌄ Markets ⌄ Sustainability ⌄ Legal ⌄ Breaking

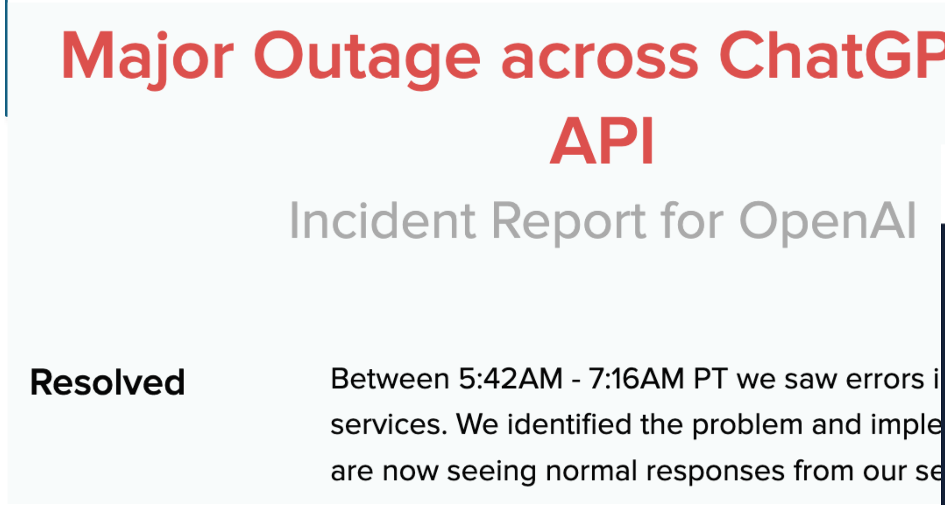
Technology

Amazon cloud services back up outage hits thousands of users



Alibaba Outage Caused by Cooling Unit

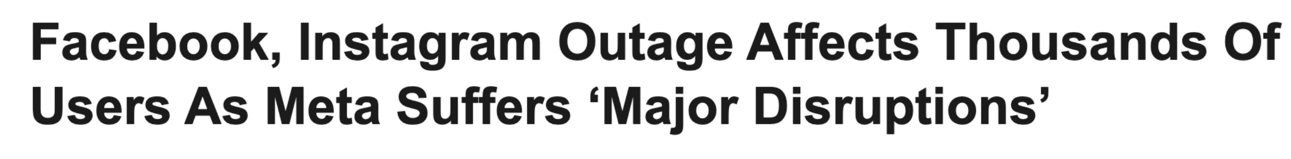
Cloud giant Alibaba's outage on Sunday was caused by a malfunctioning refrigeration unit, say company



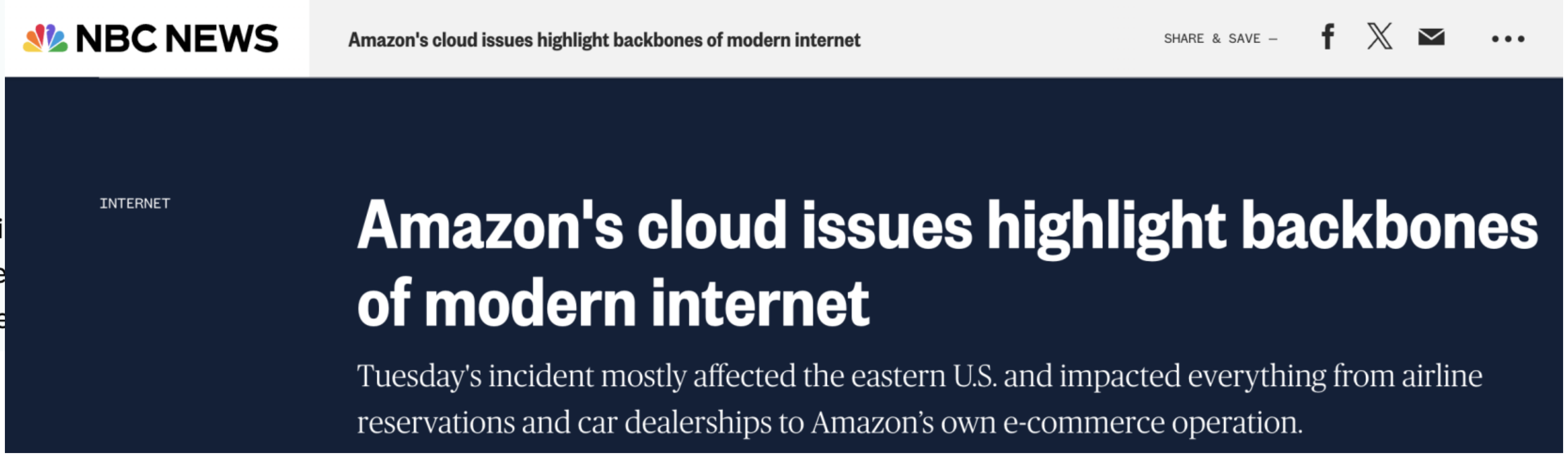
Major Outage across ChatGPT and API

Incident Report for OpenAI

Resolved Between 5:42AM - 7:16AM PT we saw errors i services. We identified the problem and imple are now seeing normal responses from our se



Facebook, Instagram Outage Affects Thousands Of Users As Meta Suffers 'Major Disruptions'



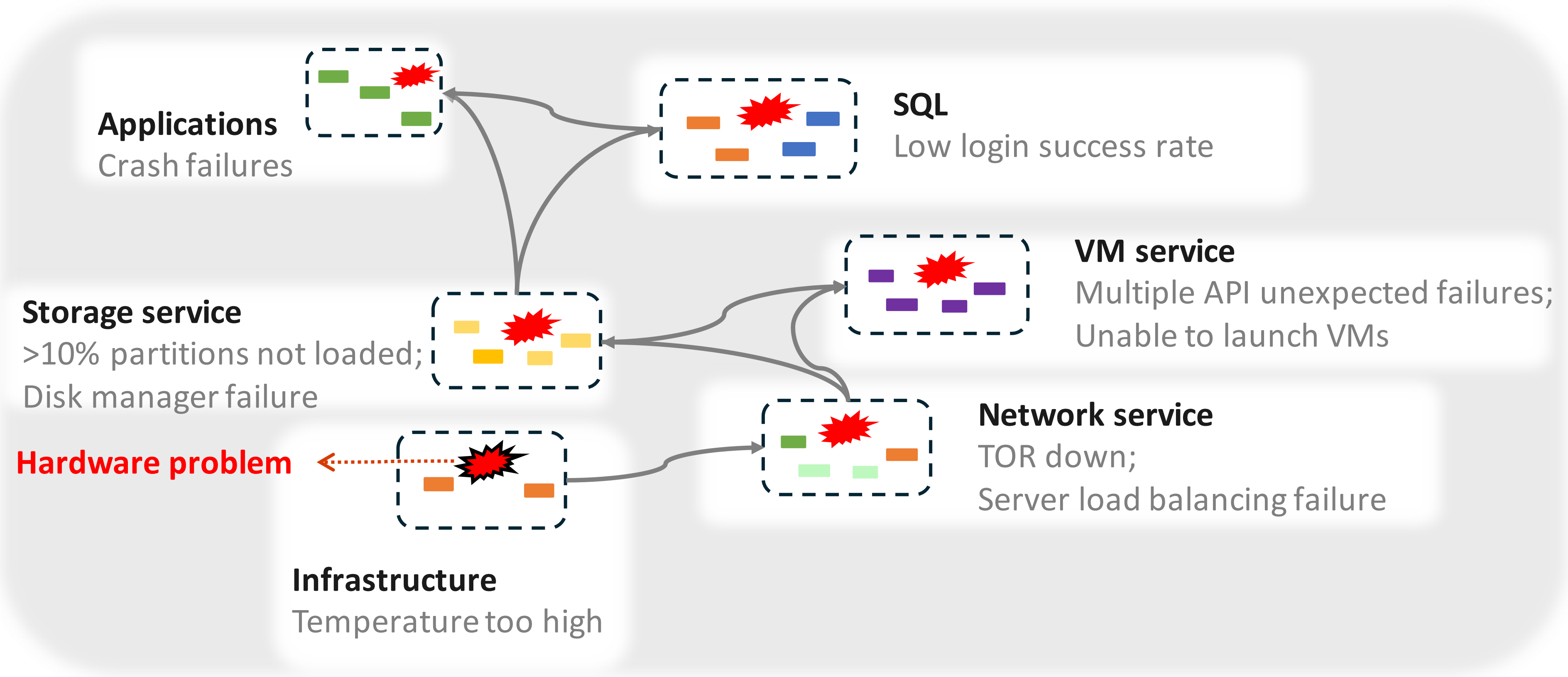
NBC NEWS Amazon's cloud issues highlight backbones of modern internet SHARE & SAVE –

INTERNET

Amazon's cloud issues highlight backbones of modern internet

Tuesday's incident mostly affected the eastern U.S. and impacted everything from airline reservations and car dealerships to Amazon's own e-commerce operation.

Incident Root Cause Analysis (RCA)



Applications
Crash failures

SQL
Low login success rate

VM service
Multiple API unexpected failures;
Unable to launch VMs

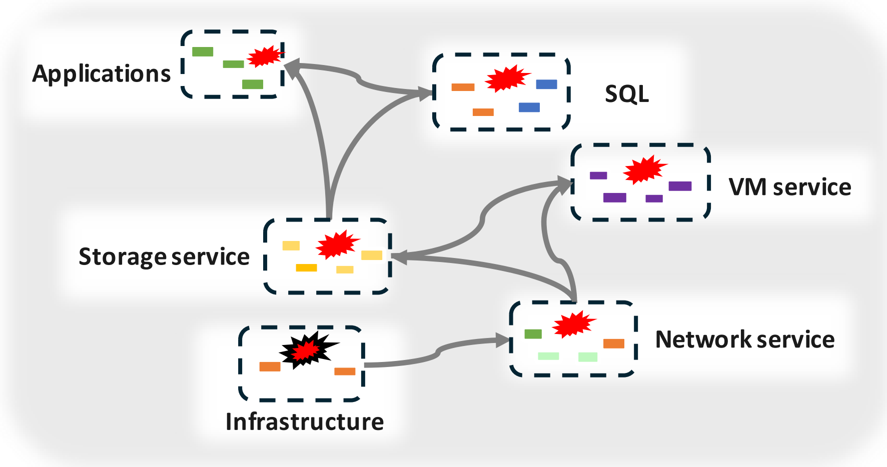
Storage service
>10% partitions not loaded;
Disk manager failure

Network service
TOR down;
Server load balancing failure

Hardware problem

Infrastructure
Temperature too high

Challenges of Incident Root Cause Analysis

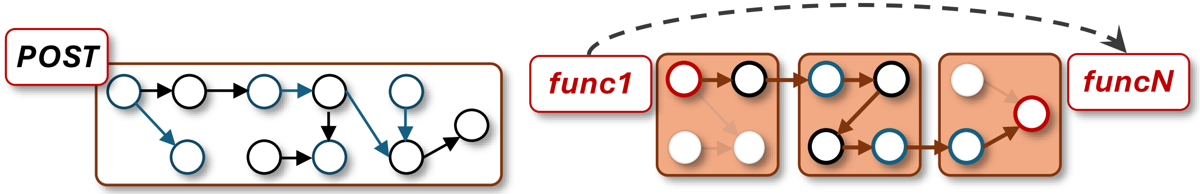


To win this war in fog, we have ...

Logs

```
07-29 19:17:57,939  – INFO [/10.10.10.01:2222]  – Received connection request /11.11.11.01:5555
07-29 19:17:57,956  – WARN [Worker: 188979561024]  – Interrupting SendWorker
07-29 19:18:01,926  – WARN [Worker: 188979561024]  – Interrupting while waiting for msg on queue
07-29 19:18:07,944  – WARN [Worker: 188979561024]  – Interrupting SendWorker
07-29 19:18:07,958  – WARN [Worker: 188979561024]  – Interrupting SendWorker
```

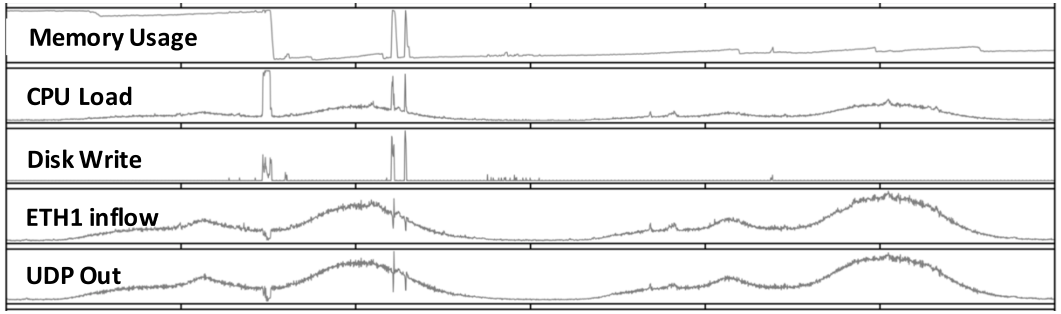
Traces



Request trace

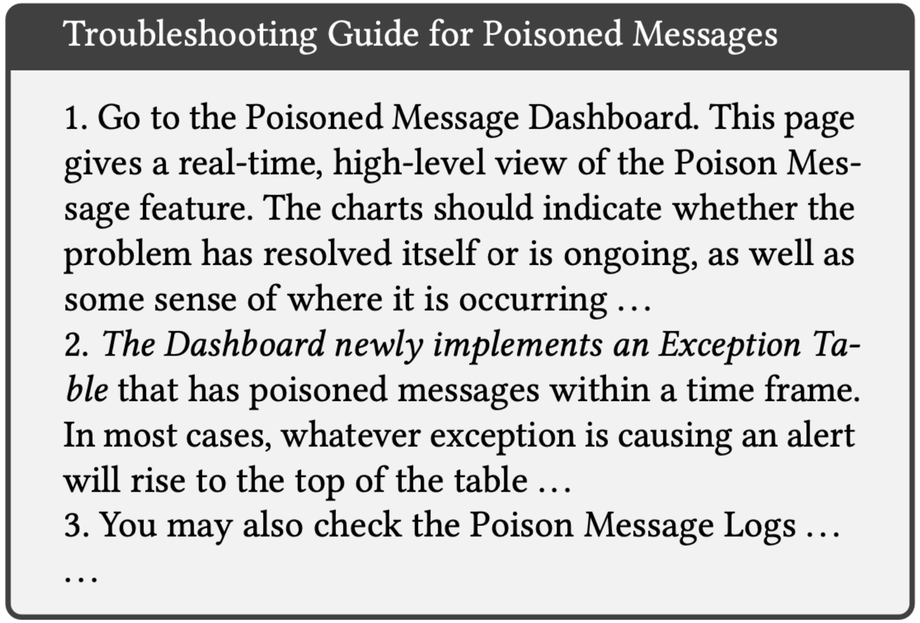
Exception trace

Metrics



Collection Challenge: The diagnostic information is hard to collect and could be **too little** or **overwhelming** for engineers.

Troubleshooting Guide is Insufficient

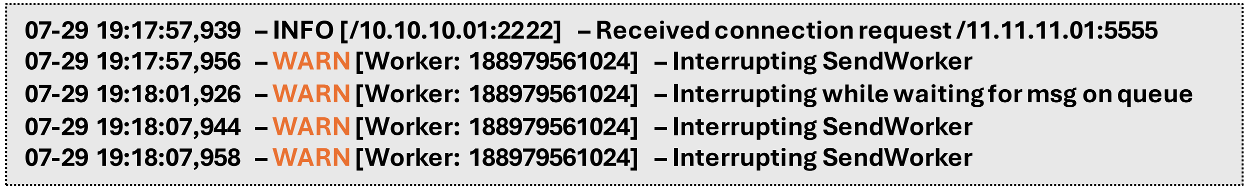


Troubleshooting Guide for Poisoned Messages

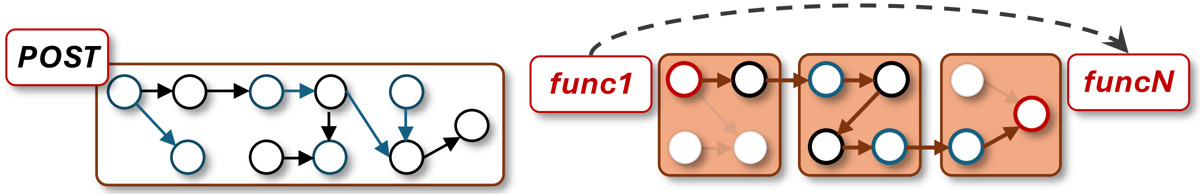
1. Go to the Poisoned Message Dashboard. This page gives a real-time, high-level view of the Poison Message feature. The charts should indicate whether the problem has resolved itself or is ongoing, as well as some sense of where it is occurring . . .
2. *The Dashboard newly implements an Exception Table* that has poisoned messages within a time frame. In most cases, whatever exception is causing an alert will rise to the top of the table . . .
3. You may also check the Poison Message Logs . . .

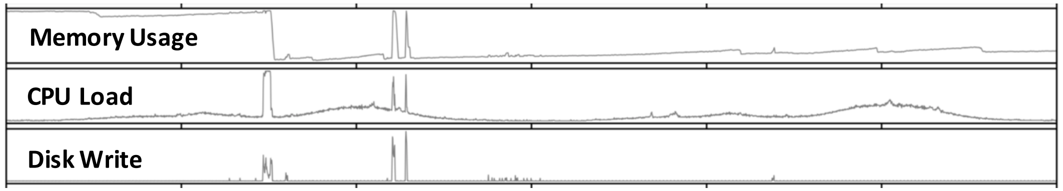
. . .

- **Wordy and hard to understand**
- Complicated to follow it step by step



```
07-29 19:17:57,939  – INFO [/10.10.10.01:2222]  – Received connection request /11.11.11.01:5555
07-29 19:17:57,956  – WARN [Worker: 188979561024]  – Interrupting SendWorker
07-29 19:18:01,926  – WARN [Worker: 188979561024]  – Interrupting while waiting for msg on queue
07-29 19:18:07,944  – WARN [Worker: 188979561024]  – Interrupting SendWorker
07-29 19:18:07,958  – WARN [Worker: 188979561024]  – Interrupting SendWorker
```

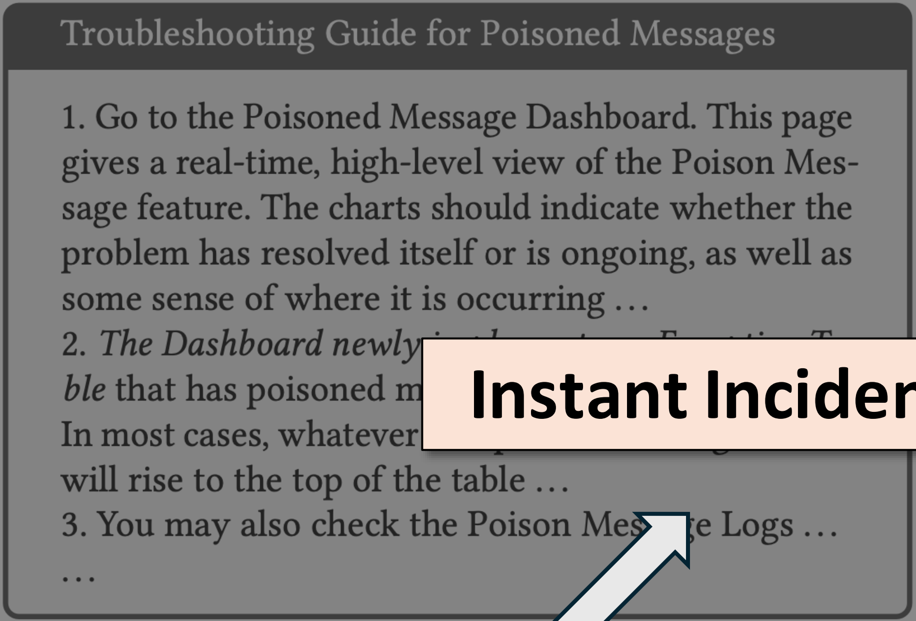




<span style="color:red">Collection Challenge</span>: The diagnostic information is hard to collect and could be too little or overwhelming for engineers.

Analysis Challenge: It is **time-consuming** for engineers to **analyze** and **interpret** the information.

Troubleshooting Guide is Insufficient

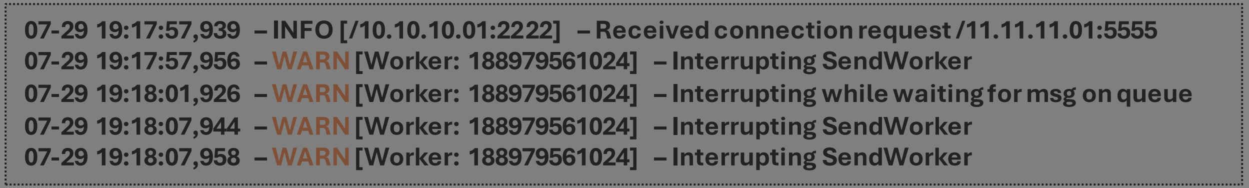


Troubleshooting Guide for Poisoned Messages

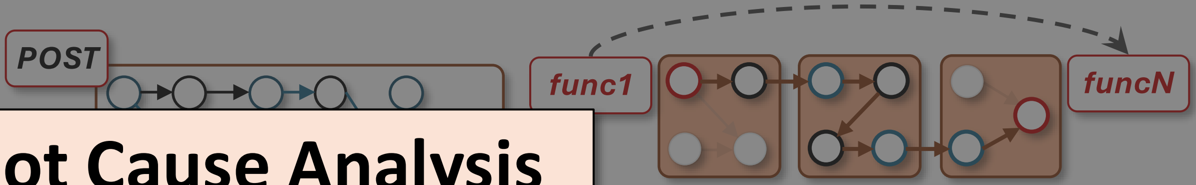
1. Go to the Poisoned Message Dashboard. This page gives a real-time, high-level view of the Poison Message feature. The charts should indicate whether the problem has resolved itself or is ongoing, as well as some sense of where it is occurring …
2. *The Dashboard newly* *ble* that has poisoned m In most cases, whatever will rise to the top of the table …
3. You may also check the Poison Mes e Logs …

…

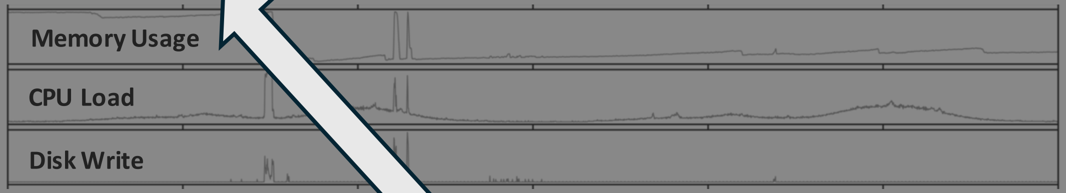
- **Wordy and hard to understand**
- Complicated to follow it step by step



```
07-29 19:17:57,939  – INFO [/10.10.10.01:2222]  – Received connection request /11.11.11.01:5555
07-29 19:17:57,956  – WARN [Worker: 188979561024]  – Interrupting SendWorker
07-29 19:18:01,926  – WARN [Worker: 188979561024]  – Interrupting while waiting for msg on queue
07-29 19:18:07,944  – WARN [Worker: 188979561024]  – Interrupting SendWorker
07-29 19:18:07,958  – WARN [Worker: 188979561024]  – Interrupting SendWorker
```



Instant Incident Root Cause Analysis



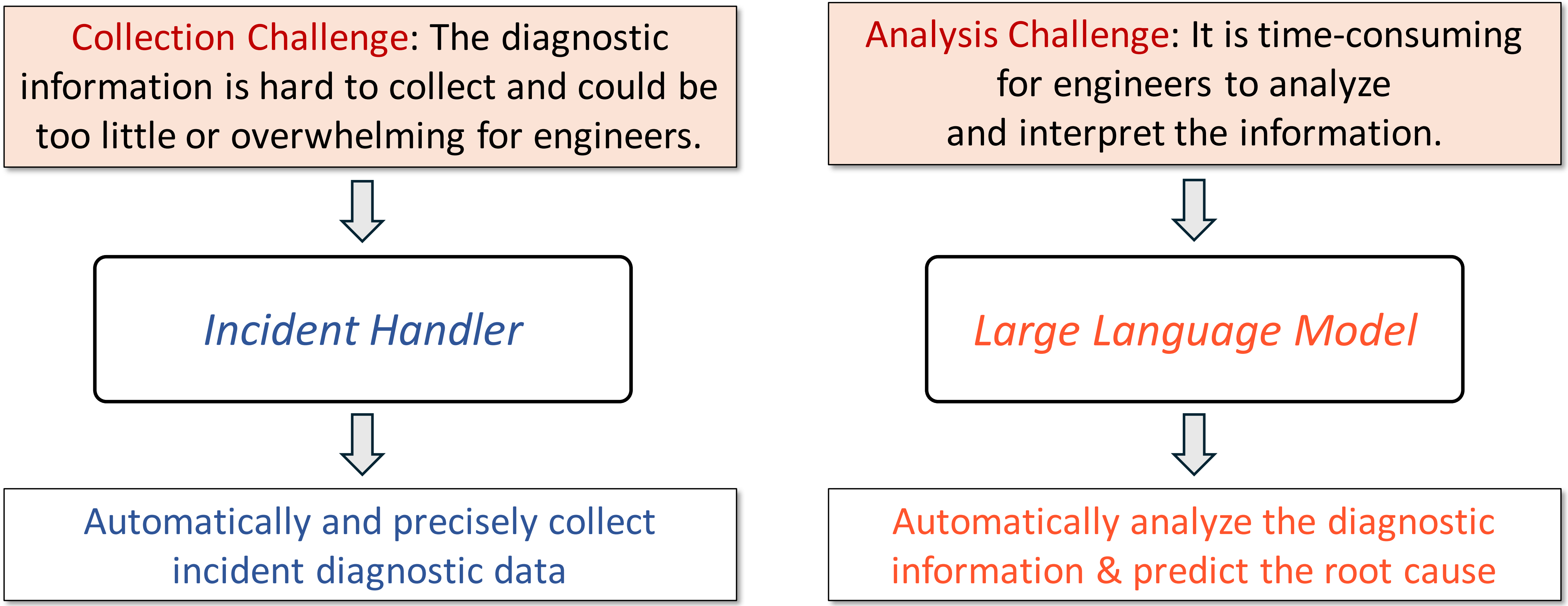
Collection Challenge: The diagnostic information is hard to collect and could be too little or overwhelming for engineers.

Analysis Challenge: It is time-consuming for engineers to analyze and interpret the information.

Contributions

- A study of the *production incidents* from a Microsoft email service
  - Derive insights on how to do effective root cause analysis
- RCACOPILOT, *an automated end-to-end on-call system* for cloud incident root cause analysis
  - Incident-specific automatic workflows for efficient data collection
  - Integration of LLMs to predict root cause categories with explanations
- Production deployment of RCACOPILOT within Microsoft

Goals of RCACopilot



Collection Challenge: The diagnostic information is hard to collect and could be too little or overwhelming for engineers.

Incident Handler

Automatically and precisely collect incident diagnostic data

Analysis Challenge: It is time-consuming for engineers to analyze and interpret the information.

Large Language Model

Automatically analyze the diagnostic information & predict the root cause

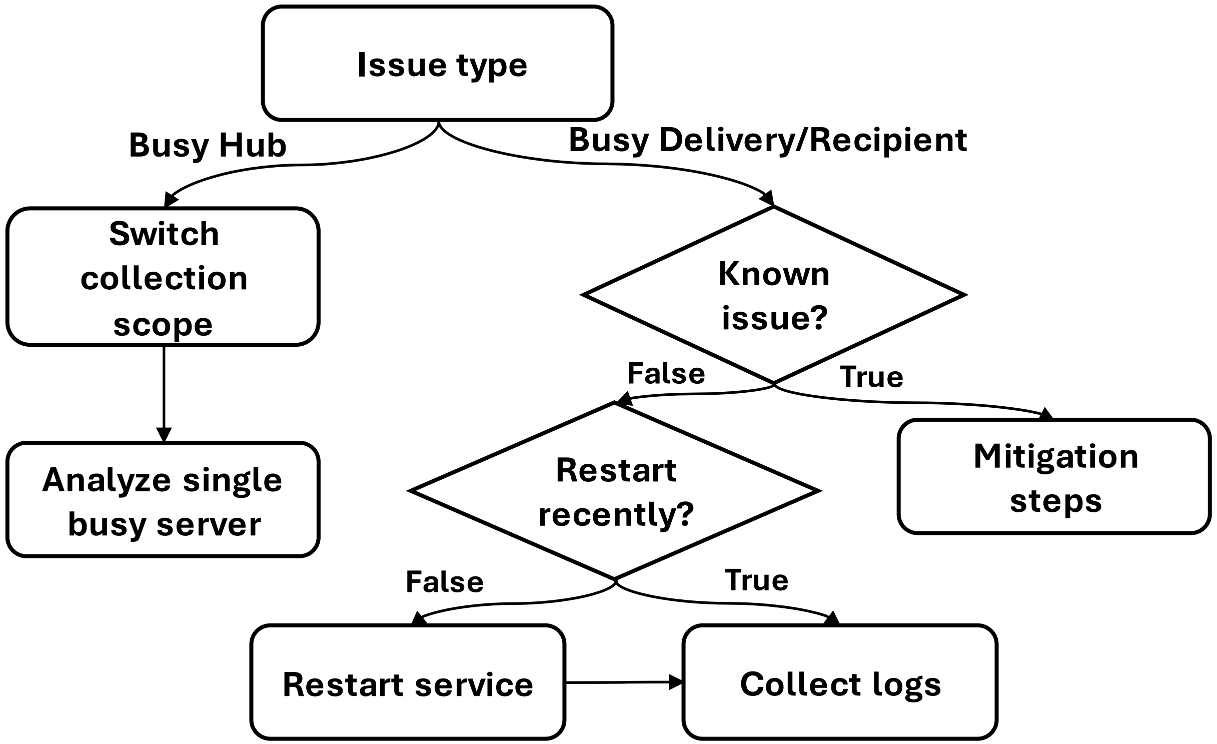
Automatic Diagnostic Information Collection

Diagnostic information collection is a resemble of a decision tree

Implemented by the incident handler of RCACOPILOT

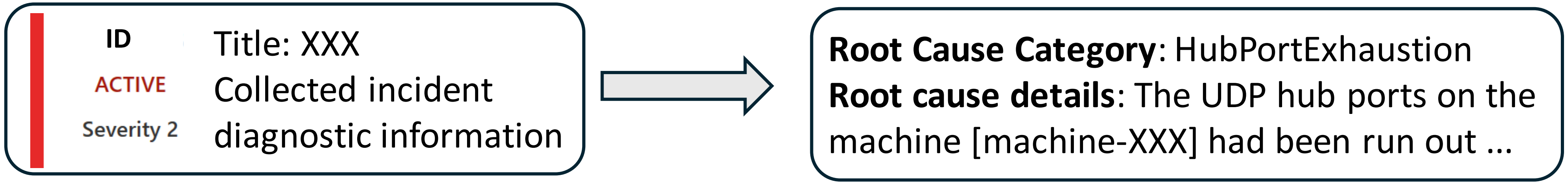
RCACOPILOT will:

- match the corresponding handler
- execute the handler
- output diagnostic information



Root Cause Prediction with LLMs

- Automatic few-shots chain-of-thoughts (CoT) prompt construction
- Root cause category prediction and explanation



ID

ACTIVE

Severity 2

Title: XXX
Collected incident diagnostic information

Root Cause Category: HubPortExhaustion
Root cause details: The UDP hub ports on the machine [machine-XXX] had been run out ...

Few-shots Chain-of-Thoughts (CoT) Prompting

Few-shots CoT:

- A few demonstrations: **historical incidents**
  - Question (Q): diagnostic information
  - Reasoning/Answer (R/A): root cause category label
- Test Question: **incoming incident's diagnostic information**

Root Cause Prediction with LLM

- Automatic few-shots chain-of-thoughts (CoT) prompt construction
- Root cause category prediction and explanation

The collected incident information cannot fit into the prompt directly:

- Long diagnostic information
- Hundreds of root cause categories
- Token limit of Large Language Models

A single incident information could contain more than 1000 tokens.

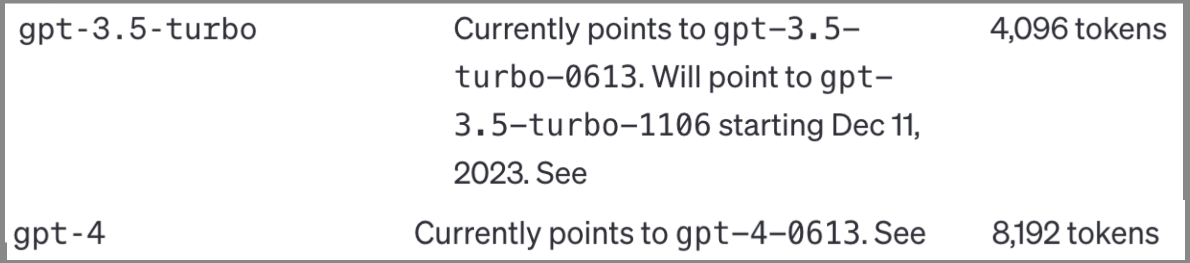
ion

on the

chine [machine-XXX] had been run out ...

Solution:

- **Similar incident retrieval**
- **Incident summarization**



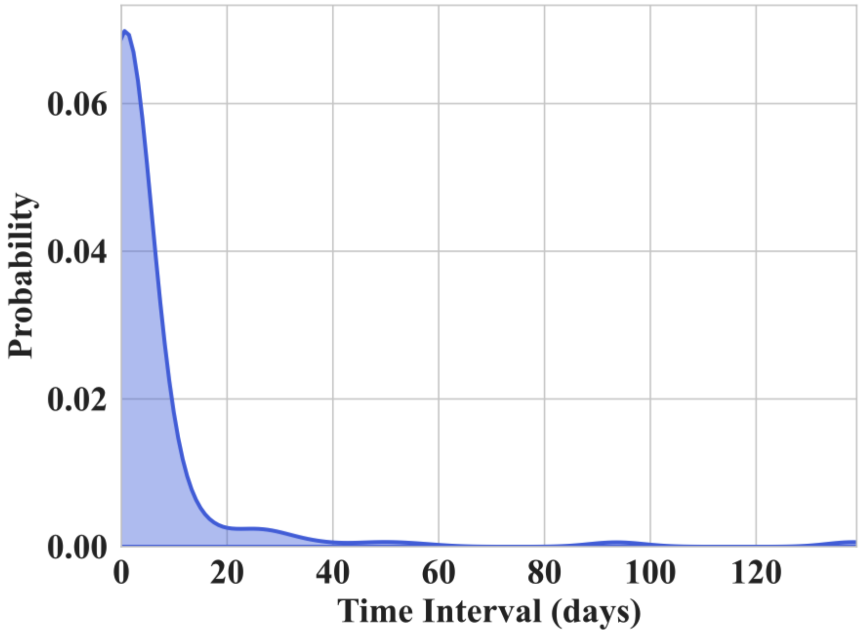
`gpt-3.5-turbo`	Currently points to `gpt-3.5-turbo-0613`. Will point to `gpt-3.5-turbo-1106` starting Dec 11, 2023. See	4,096 tokens
`gpt-4`	Currently points to `gpt-4-0613`. See	8,192 tokens

Similar Incident Retrieval

- On-call engineers refer to historical incidents – Provide examples for LLM

How to measure the similarity?

- Study insight: incidents stemming from the same root cause often recur within a short period – **Time locality**



Most recurring incidents (93.8%) tend to reappear within 20 days.

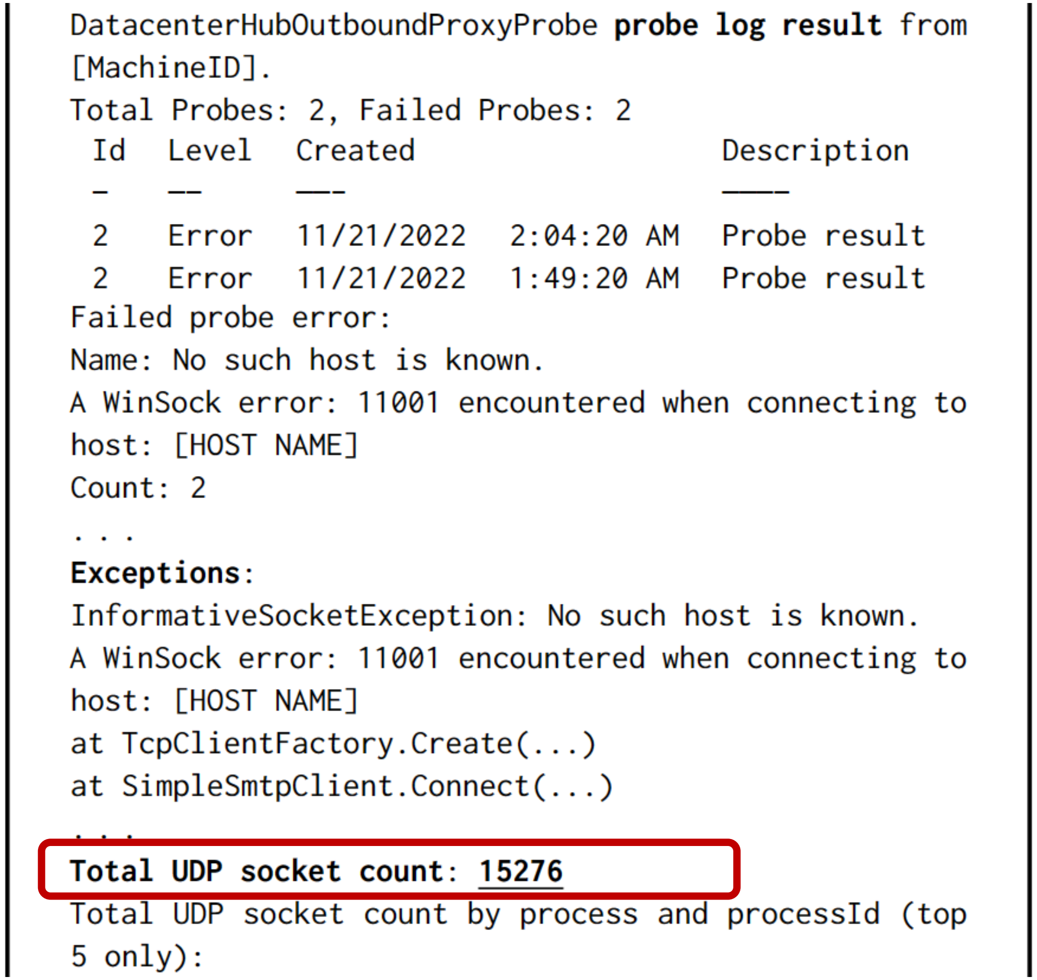
When retrieving:

- Embedding vector distance between diagnostic informtion
- Temporal distance between incidents

Incident Summarization

Original diagnostic data collected by incident handler

1000+ tokens



```
DatacenterHubOutboundProxyProbe probe log result from
[MachineID].
Total Probes: 2, Failed Probes: 2
 Id  Level  Created                     Description
 -   --     ---                         ----
 2   Error  11/21/2022   2:04:20 AM    Probe result
 2   Error  11/21/2022   1:49:20 AM    Probe result
Failed probe error:
Name: No such host is known.
A WinSock error: 11001 encountered when connecting to
host: [HOST NAME]
Count: 2
. . .
Exceptions:
InformativeSocketException: No such host is known.
A WinSock error: 11001 encountered when connecting to
host: [HOST NAME]
at TcpClientFactory.Create(...)
at SimpleSmtpClient.Connect(...)
. . .
Total UDP socket count: 15276
Total UDP socket count by process and processId (top
5 only):
```

Prompt used in summarization:

Please summarize the above diagnostic information. The summary results should be about 120 words ...

RCACopilot summary result:

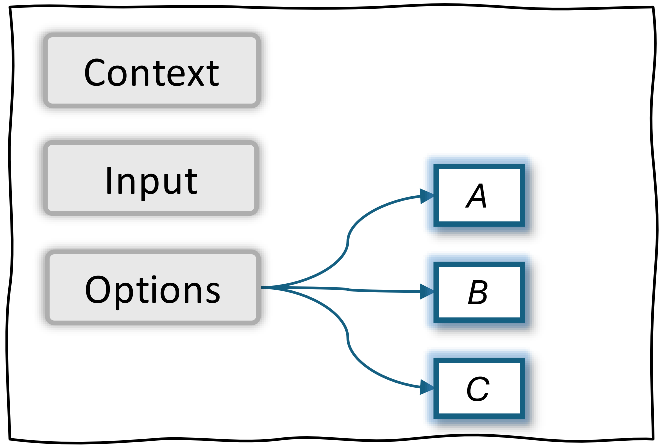
The Datacenter Hub Probe has failed twice on the backend machine, ...

This error was encountered while attempting to connect to the host ...

The total UDP socket count is 15276, with the majority being used by the serviceX.exe process.

... ...

Automatic Chain-of-Thoughts Prompting



Context: The following description shows ...
Please select ... the same root cause and give explanation ...

Input: The DatacenterHubOutboundProxyProbe probe result from [BackEndMachine] is a failure...

Options:

- A: ***Label***: Delivery hang.
  Summary: There are 62 managed threads in process [MSExchangeDelivery]...
- B: ***Label***: Code regression.
  Summary: The DatacenterHubOutboundProxyProbe probe failed with ...
- C: ***Label***: None

Evaluation

- Is RCACOPILOT effective and efficient as an on-call system?
- How different components of RCACOPILOT facilitate its diagnosis and prediction?

Evaluation Results

RCACopilot achieves 0.766 F1-score when predicting the root causes.

Method	F1-score		Prediction Stage Time (sec.)	
	Micro	Macro	Train.	Infer.
XGBoost	0.022	0.009	11.581	1.211
Fine-tune GPT	0.103	0.144	3192	4.262
GPT-4 Prompt	0.026	0.004	-	3.251
GPT-4 Embed.	0.257	0.122	1925	3.522
RCACopilot (GPT-3.5)	0.761	0.505	10.562	4.221
RCACopilot (GPT-4)	**0.766**	**0.533**	10.562	4.205

Evaluation Results

RCACOPILOT has been deployed in an email service (150 billion messages delivered daily) at Microsoft.

Data Source			F1-score	
AlertInfo	DiagnosticInfo	ActionOutput	Micro	Macro
	☑		0.689	0.510
	☑ sum.		**0.766**	**0.533**
☑			0.379	0.245
☑	☑		0.525	0.511
☑		☑	0.431	0.247
	☑	☑	0.501	0.449
☑	☑	☑	0.440	0.349

Teams using Collection Module		
Team	Exec Time (sec.)	# Handler
1	841	213
2	378	204
3	106	88
4	449	42
5	136	41

Conclusion

- A study of the *production incidents* from a Microsoft email service
    - Derive insights on how to do effective root cause analysis
- RCACOPILOT, an *automated end-to-end on-call system* for cloud incident root cause analysis
    - Incident-specific automatic workflows for efficient data collection
    - Integration of LLMs to predict root cause categories with explanations
- Production deployment of RCACOPILOT within Microsoft

Automatic Root Cause Analysis via Large Language Models for Cloud Incidents