Project 1 (EDA)
Exploratory Data Analysis

Ehu Shubham Shaw
Roee Shrug
September 12, 2024

Task 1 :

Motivation : Coming from a DevOps background, I understand the critical importance of extracting and accessing crucial data, especially when the pressure is on in a production environment. When the stakes are high, and every second counts, being two steps ahead is essential to keep applications running smoothly. In such scenarios, logs are invaluable—they play a vital role not only for DevOps engineers but also for software engineers, providing key insights for troubleshooting, monitoring and performance enhancement or getting system up and running from a failure state.

My Log data Analysis contains of:

- Zookeeper_2k.log,
- Mac_2k.log,
- Hadoop_2k.log

url : https://github.com/logpai/loghub/tree/master/Hadoop

This dataset represents Log Data from different application like Hadoop, Mac and Zookeeper.

1. Zookeeper logs.

In Apache Kafka, Zookeeper is used to manage and coordinate Kafka brokers (servers) in a distributed environment. It helps track of brokers, leader election and also contain information about topics, partitions, and the brokers.These logs usually contains Error and Debug Logs, Audit Logs, Transaction Logs.

- These logs record error messages, warnings, and debugging information, such as connection issues, client requests, and failures in the system(eg:- Out of memory issues). This information is crucial for diagnosing problems, monitoring performance of kafka speed, and ensuring the health of the Zookeeper.
- These logs record user actions and access details, which are useful for security audits. Also these logs are crucial for Data Recovery as transaction can be find in logs for recovery.

They usually occur in these form : Errors, Warnings (System events)

2. Hadoop Logs.

Hadoop is an open source framework based on Java that manages the storage and processing of large amounts of data for applications. Hadoop logs contain detailed information about the activities, operations and errors that occur within a Hadoop system. Maybe while preforming a task. And it may contain information from Security and Access Logs, Resource Usage Information, Error Messages and Stack Traces, Configuration Information.

- While troubleshooting or debugging during your Hadoop job failure or to make it run correctly its important for root cause analysis and diagnostics to fix the issue. node failure can be detected early through logs in hadoop.

- It also helps in audit and security which are crucial for companies for tracking access and operations performed by users, it also helps in resource management of resource and prevent overloading.

They usually occur in these form : Errors, Warnings.

3. Mac Logs.

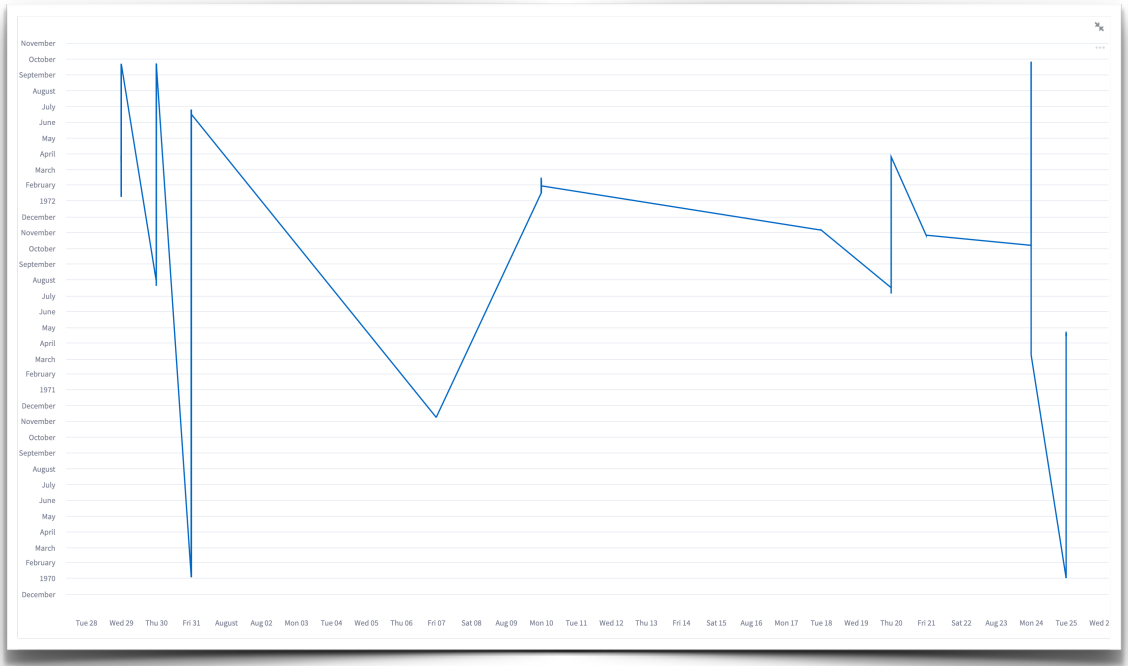
Mac logs are useful for understanding what is happening on a Mac computer, diagnosing problems, and ensuring the system is running smoothly. Also these log file contains various entries related to system events, kernel messages, application logs, and network activities.

- While troubleshooting or debugging we can catch systems unexpected behavior or some error such as hardware issues (eg. charging port), application crash. Also diagnose problems with network in Mac (eg. Wifi, Hotspot).
- Security and Auditing can also be tracked for user activity through access logs, and detect potential security incidents, Monitor power management, hardware status (eg : keyboard, screen etc.).

They usually Occur in this form : Debug,Warning, Warn, Err, Failed, Crash, Critical, Critical Error, Alert,Emergency.

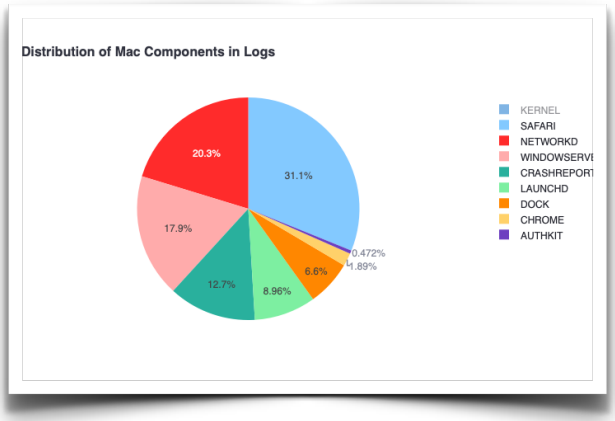
Task 2 :

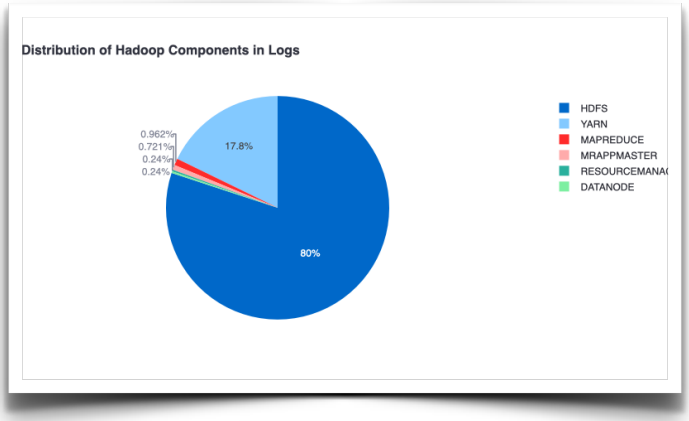
- Log Entries Over Time.

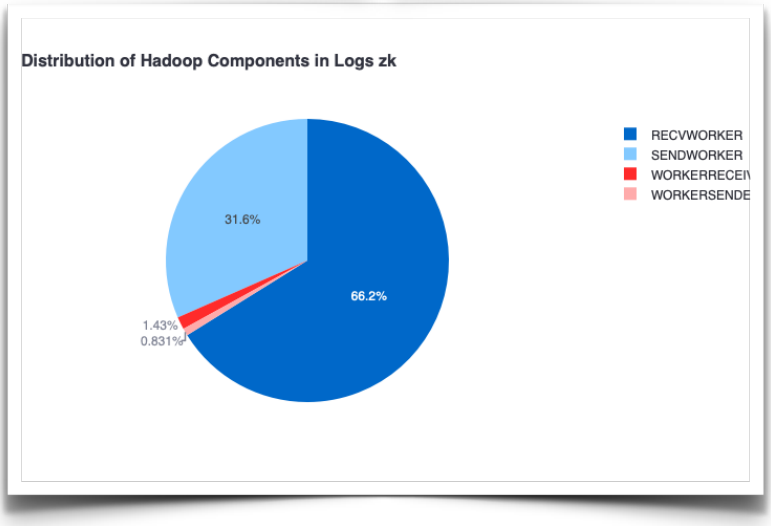


Summery : This log's Entering over time in a zookeeper cluster can indicate insights. It can indicate messaging activity preformed in the time frame which indicates dependency of the individual standalone instance/over all the cluster health on the server for the day, network latency/memory issues can be monitored as if any latency in message can lead to use of another instance which means the individual instance of zookeeper was not in use.

- Mac, Hadoop & Zookeeper Component Distribution.







Summery :

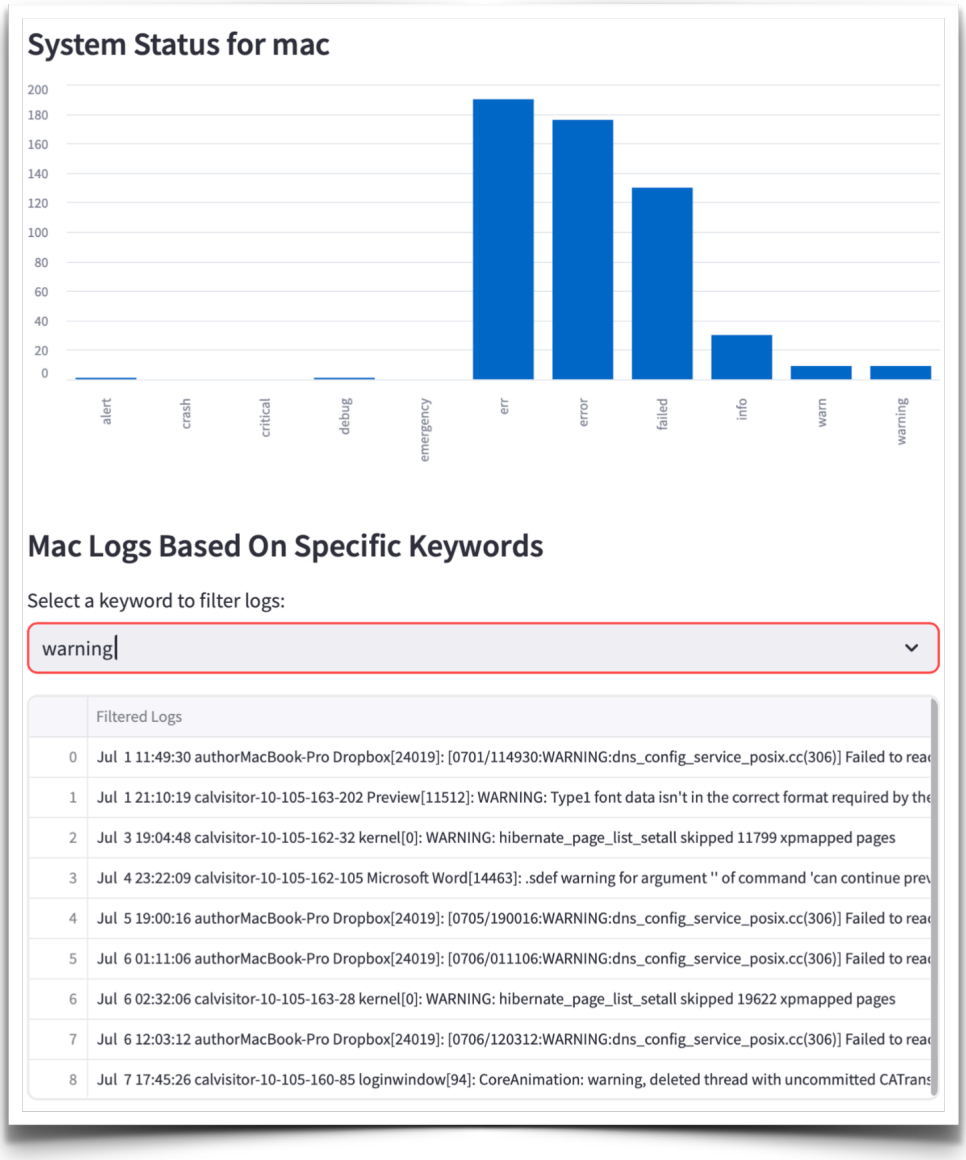
In a Hadoop ecosystem, various components responsible for different tasks, for now we have "***MRAppMaster***" which manages the lifecycle of map reduce jobs. A high percent on pie indicates it indicate lot of job life cycle management is happening which might lead to scheduling/execution of job. Similarly "**YARN**" indicates potential config issue or high resource management activity. Eg: CPU or disk usage. Potential issues related to data replication, corruption in "**HDFS"** can be seen in pie chart as pdfs is responsible for data storage and retrieval operations, hight percentage on "***ResourceManager and DataNode"*** could indicate request data node operations which might be because of storage issues or network problems

In Mac pie chart the word **kernel** occurred 775 times the kernel component is highly active, suggesting frequent logging related to system operations, errors. High percent in pie for "**networkd**" indicates

moderate activity or network-related events or issues. Whereas **mDNSResponder** potential issues with network discovery issues, The **launchd** process is mentioned less frequently, indicating limited logging activity for system launches. Minimal mentions of syslog suggest fewer logged events via **syslog logger**. **ReportCrash** indicate crash in application.

Zookeeper pie chart indicates distribution of worker activity types, a pie chart helps in visualizing how the workload is distributed among different worker types as like **RecvWorker** responsible for receiving data/messages from other nodes or clients Hight percent for this indicates high messaging activity and **SendWorker** for sending messages to other clients. **WorkerReceiver** is a worker processing incoming messages, **WorkerSender** is a worker responsible for sending out messages or data this individual percentage in pie indicates activity or logging frequency/count. Analyzing the distribution of these worker types means calculating how often each worker type appears in the log file, which provides a measure of how much activity or workload is handled by each type.

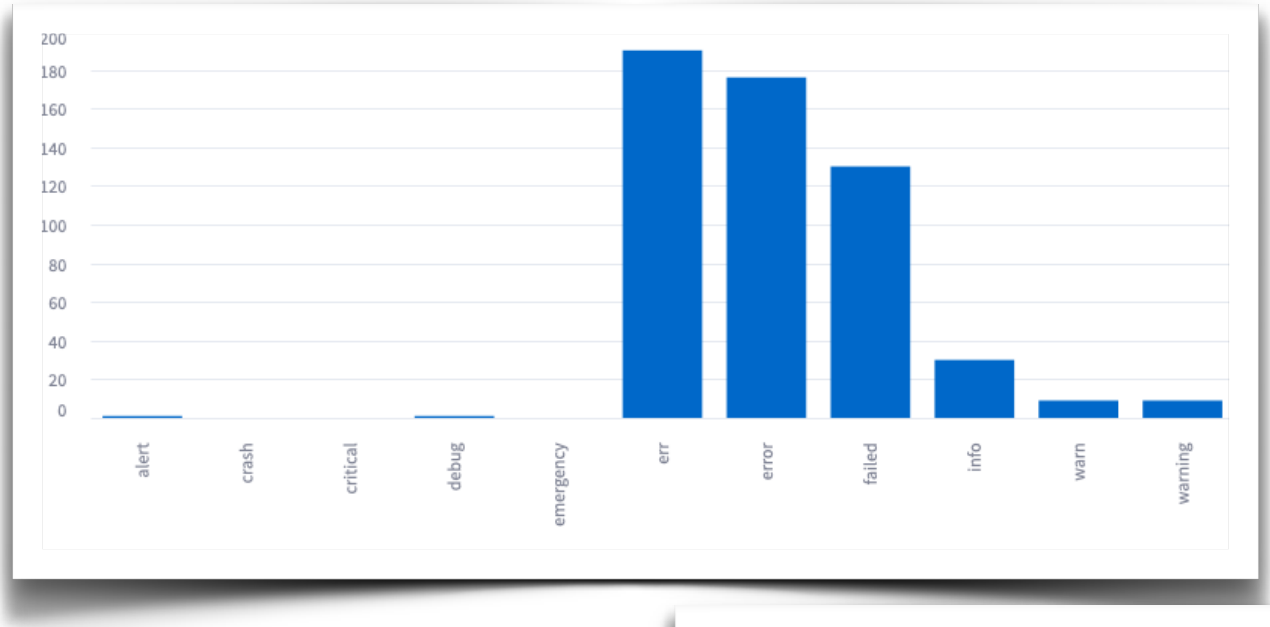
- Mac Logs Based On Specific Keywords.

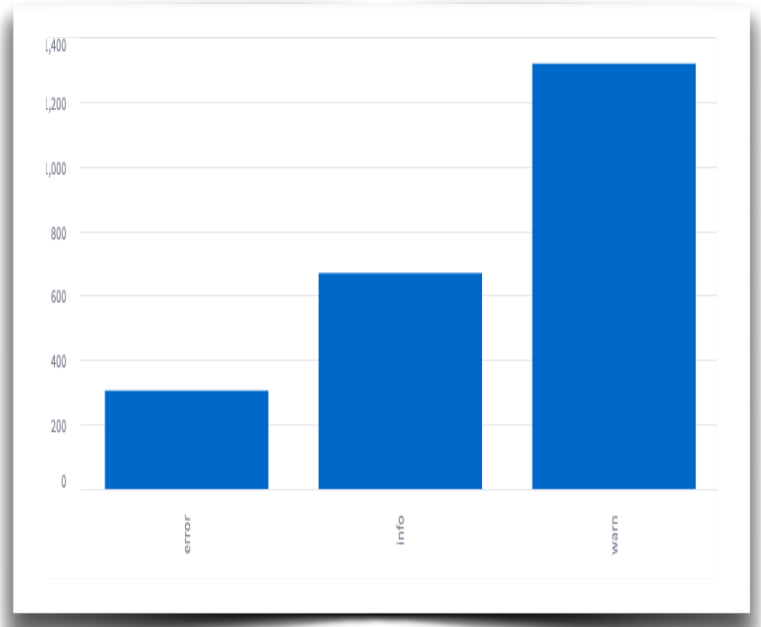


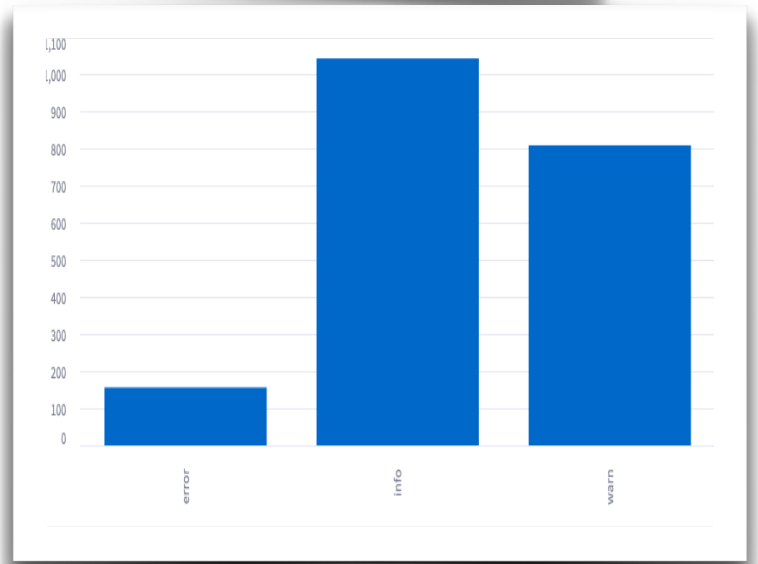
Summery : Having logs in the data view itself is a blessing it saves the hassle to go in application log and searching for error type, if we have the logs in hand in form of list it saves a lot of hassle.

- ***General System Status for Mac, Hadoop & Zookeeper.***

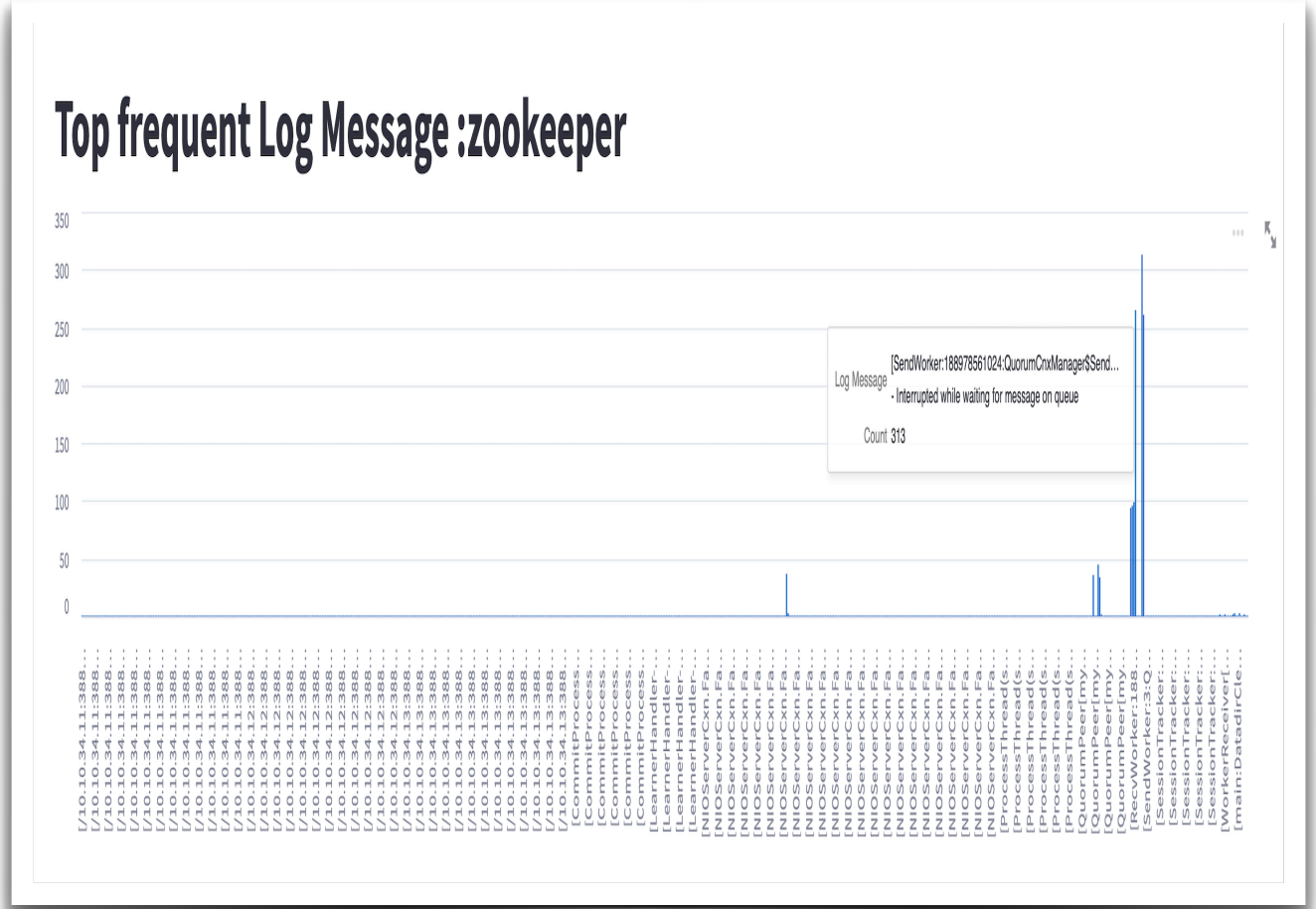
Summery : A comprehensive overview of the entire system's performance can provide invaluable insights into whether an application is running optimally or if there is a noticeable increase in the number of errors. By closely monitoring key components and their activity counts, it becomes easier to identify when these metrics are approaching critical thresholds. This proactive monitoring allows for the early detection and resolution of potential issues before they escalate into more severe problems, thereby preventing potential damage and ensuring smooth and efficient application performance.







- ***Top frequent Log Message : zookeeper***



Summery :

This visualization serves as a crucial tool for identifying potential issues in the system. For instance, if the message in the log states, “Login Network / Network service didn’t start properly” or mentions being “interrupted while waiting for a message in the queue,” it highlights a network service startup issue or a message handling bottleneck. This visualization allows you to take a holistic view of the entire log data, making it easier to detect such problems across different instances. By visualizing these patterns, system administrators can quickly pinpoint problematic areas and address them proactively, preventing disruptions or downtime and ensuring smooth system operations without causing further impact.

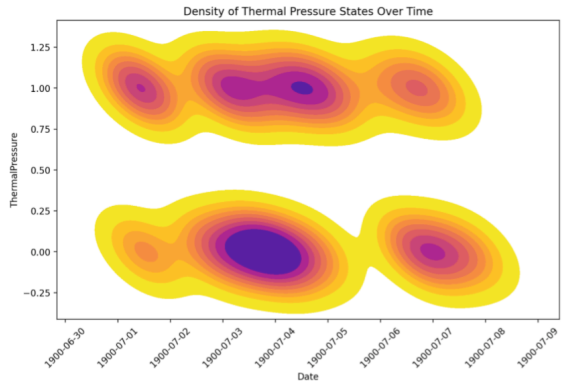
Task 3 :

- **Extracted Thermal and Memory Pressure Data over time of mac**

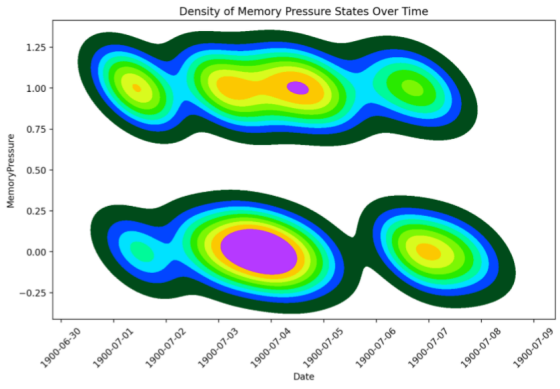
Extracted Thermal and Memory Pressure Data over time

	Timestamp	ThermalPressure	MemoryPressure
0	1900-07-01 09:01:05	1	1
1	1900-07-01 09:28:50	1	1
2	1900-07-01 09:41:54	0	0
3	1900-07-01 09:44:32	1	1
4	1900-07-01 13:41:58	1	1
5	1900-07-01 14:38:45	0	0
6	1900-07-02 18:09:15	0	0
7	1900-07-02 19:21:05	1	1
8	1900-07-02 19:48:30	0	0
9	1900-07-03 01:06:48	1	1

Thermal Pressure Time

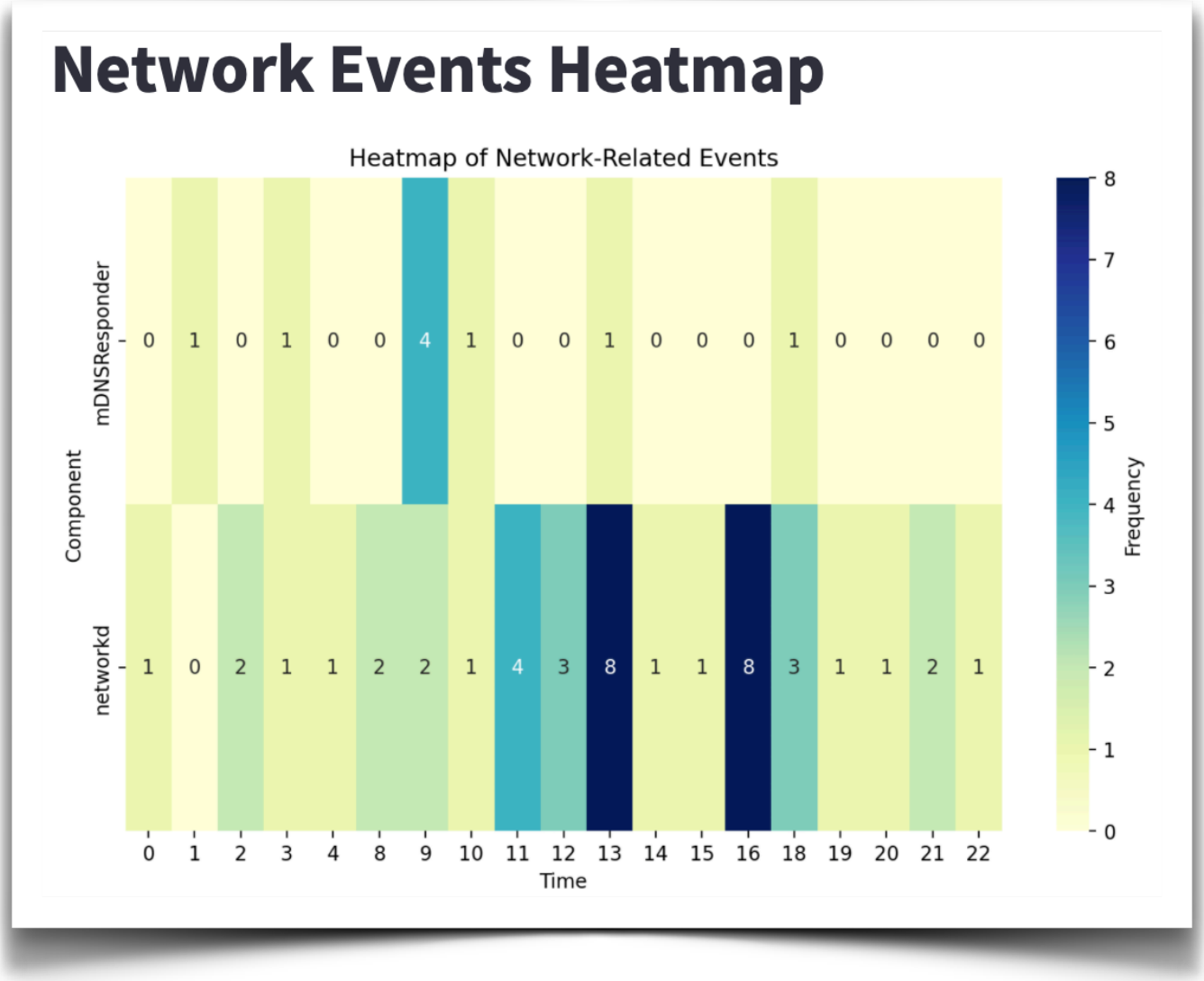


Memory Pressure over Time



Summery : This graph provides valuable insights into specific points in time when memory pressure increases, allowing us to identify patterns of resource strain. The accompanying heat maps further confirm these findings by highlighting areas where thermal pressure builds up, potentially leading to thermal shutdowns or indicating faulty hardware components. By visualizing these correlations, we can not only detect when and where these issues occur but also pinpoint the applications responsible for excessive resource consumption. This enables targeted intervention to address the root causes, optimize resource usage, and prevent hardware failures or performance degradation.

- **Network Events occurance vs time Heatmap.**



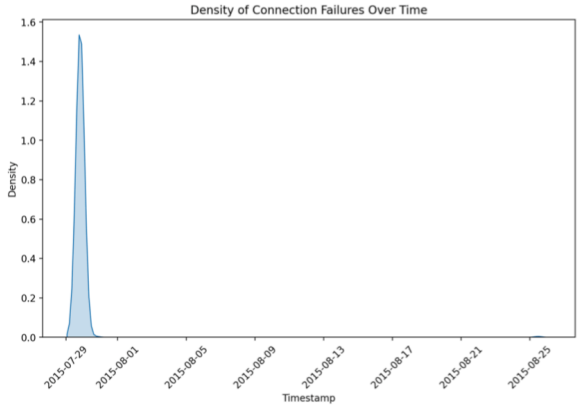
Summery : Each row represents a network-related component such as mDNSResponder, networkd, these components are responsible for various networking tasks on macOS, such as DNS resolution **mDNSResponder**, and network related issues by network configuration management **networkd**, on x column each column represents an hour of the day (00–23). This segmentation shows how the frequency of network-related events changes across different hours. The value in each cell represents the frequency of events for a given component during a specific hour the color intensity of the cell indicates the number of events (darker means more frequent). The heat-map provides a clear and concise visualization of when and how frequently different network components are generating events. This can help in identifying patterns, diagnosing network issues, and improving network stability.

- **KDE Plot for Connection and request Failures Over Time**

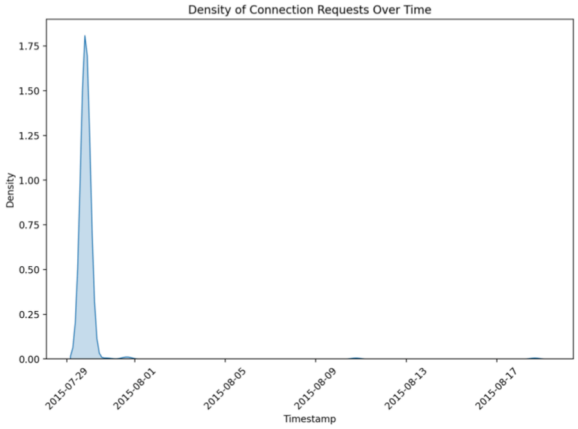
Extracted Job Duration Data

	Timestamp	EventType
0	2015-07-29 19:04:12	Connection Request
1	2015-07-29 19:13:24	Connection Failure
2	2015-07-29 19:13:24	Connection Request
3	2015-07-29 19:13:27	Connection Failure
4	2015-07-29 19:13:47	Connection Failure
5	2015-07-29 19:13:54	Connection Request
6	2015-07-29 19:13:54	Connection Failure

KDE Plot for Connection Failures Over Time

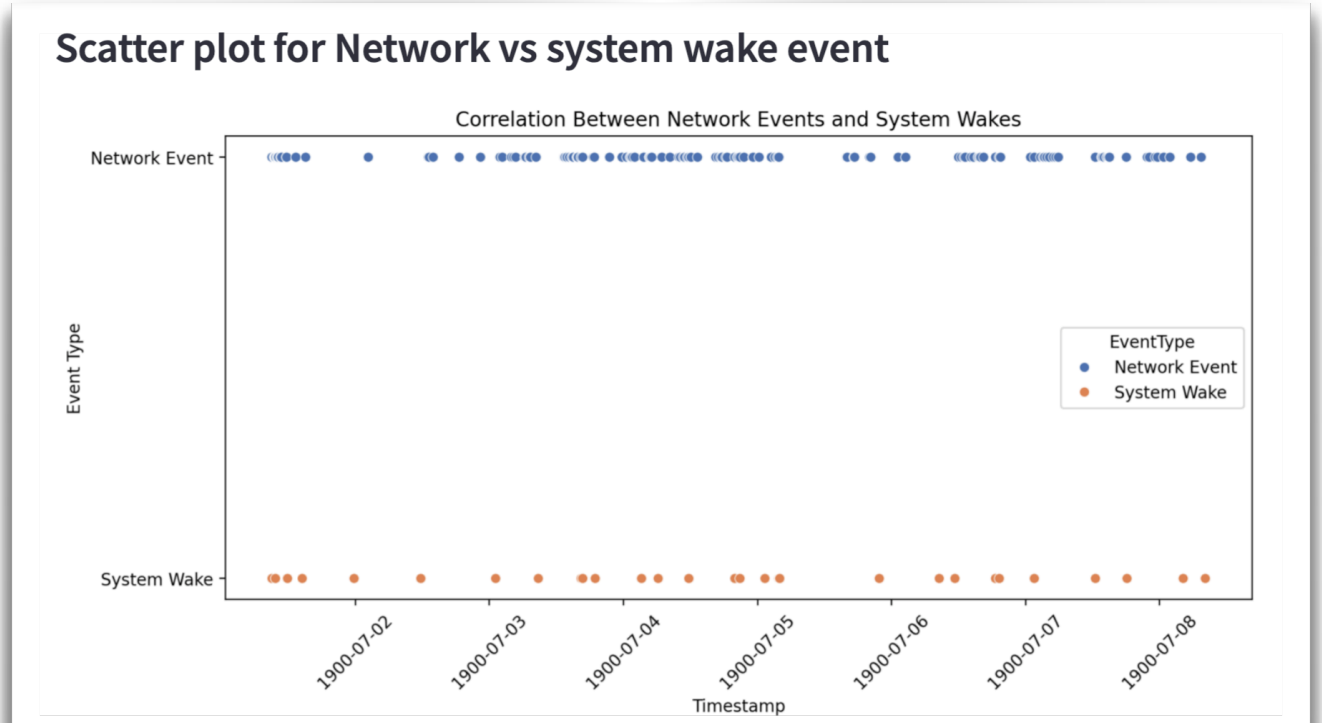


KDE Plot for Connection Requests Over Time



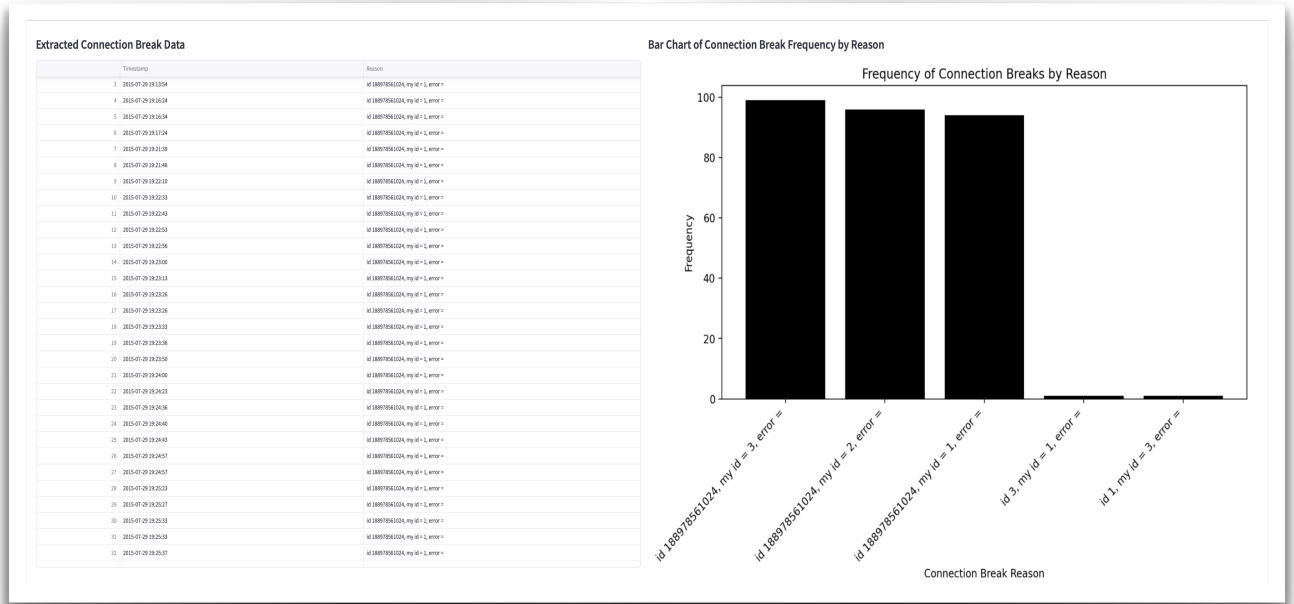
Summery : In Zookeeper, efficient connections are vital for application performance. A bar graph showing high disconnection failures may indicate slow internet or unstable networks, while many queued connection requests suggest application bugs or configuration issues. The KDE plot visualizes the density of failures over time, helping pinpoint specific days for focused log analysis and debugging, enabling quick resolution of network or application problems.

- **Scatter plot for Network vs system wake event Mac**



Summery : The scatter Plot indicates the system network usage while startup which in some case can lead to slow boot ups if self network diagnostics occur more often or if network issue is faced in application, rather than seeing the application we can make it out by this scatter plot if the network parameters were proper during boot or not.

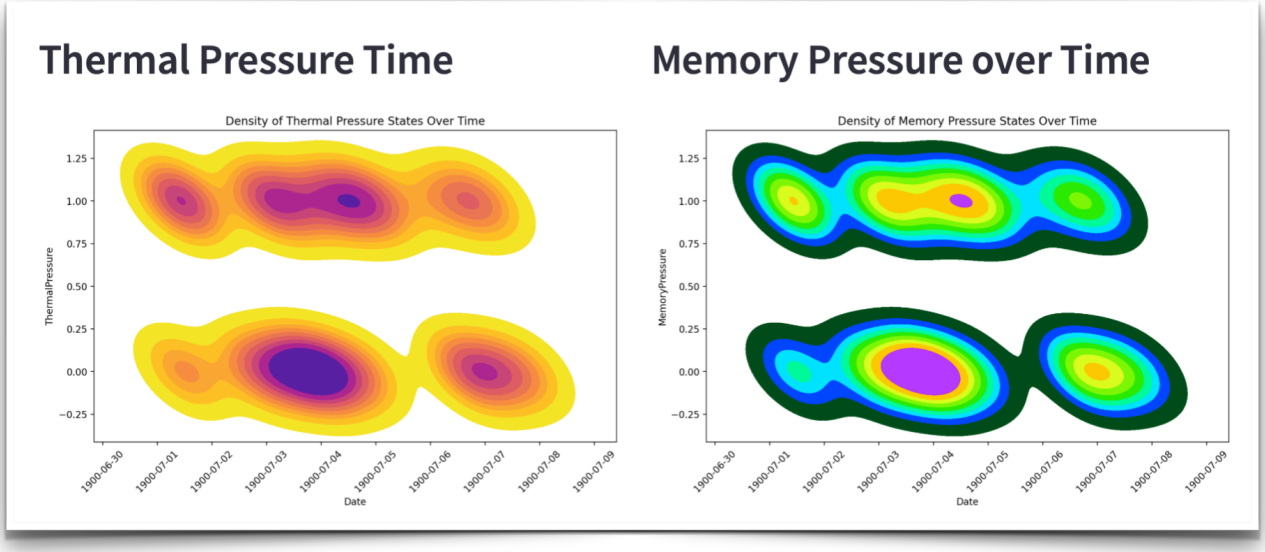
- **Bar Chart of Connection Break Frequency by Reason**



Summery : In Zookeeper, a “Connection broken” error typically indicates that a broker or schema registry is down due to memory issues or excessive RAM usage, leading to connection breaks. Alternatively, network lag can trigger an “Out of queue” error. The visualization provides a comprehensive overview, allowing you to see all issues in one place. By showing the frequency of connection breaks, it helps quickly identify and address these problems before they escalate.

Task 4 :

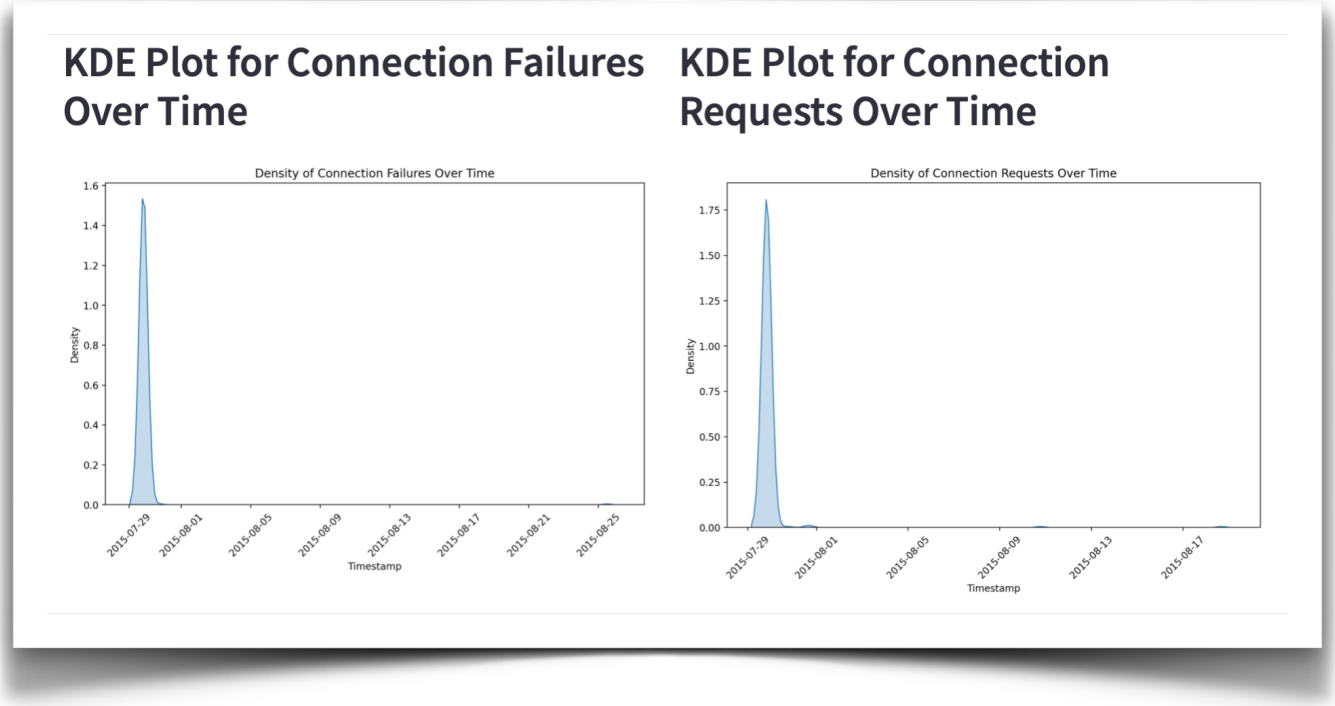
1. What Are the Common Causes for a Mac to Shutdown suddenly, Crash Applications, or Perform Slowly?



Solution *:* The “***CDScheduler***” component logs a value of 1 when a Mac overheats, indicating a problematic state. This can lead to thermal shutdowns to protect hardware. Overheating may occur due to intense CPU/GPU usage or inadequate cooling. High memory or CPU pressure is also logged with a 1 value by “***CDScheduler***”. This state can cause applications to crash or the system to slow down. Causes include too many running applications.

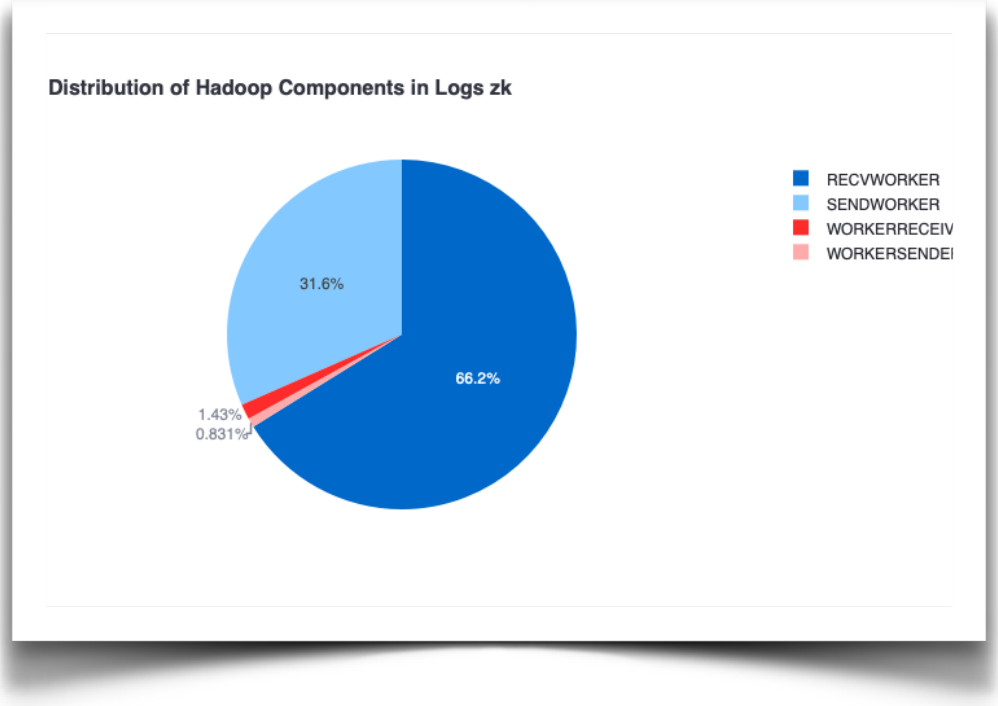
2. What do Connection Failures and Connection Requests Indicate in Log Data, and How Might These Events Impact Application Performance?

Solution: when clients trying to establish or re-establish a connection, A high frequency of connection requests could mean frequent reconnections, possibly due to unstable network conditions or server restarts or load balancing.Connection failures in logs suggest issues with maintaining a stable connection between clients and the server. That can be a issue of config or server overload or possible resource exhaust.



3. How Can I Identify if Workers in Zookeeper Are Overloaded, and What Are the Potential Impacts on System Performance?

Solution ***:*** when

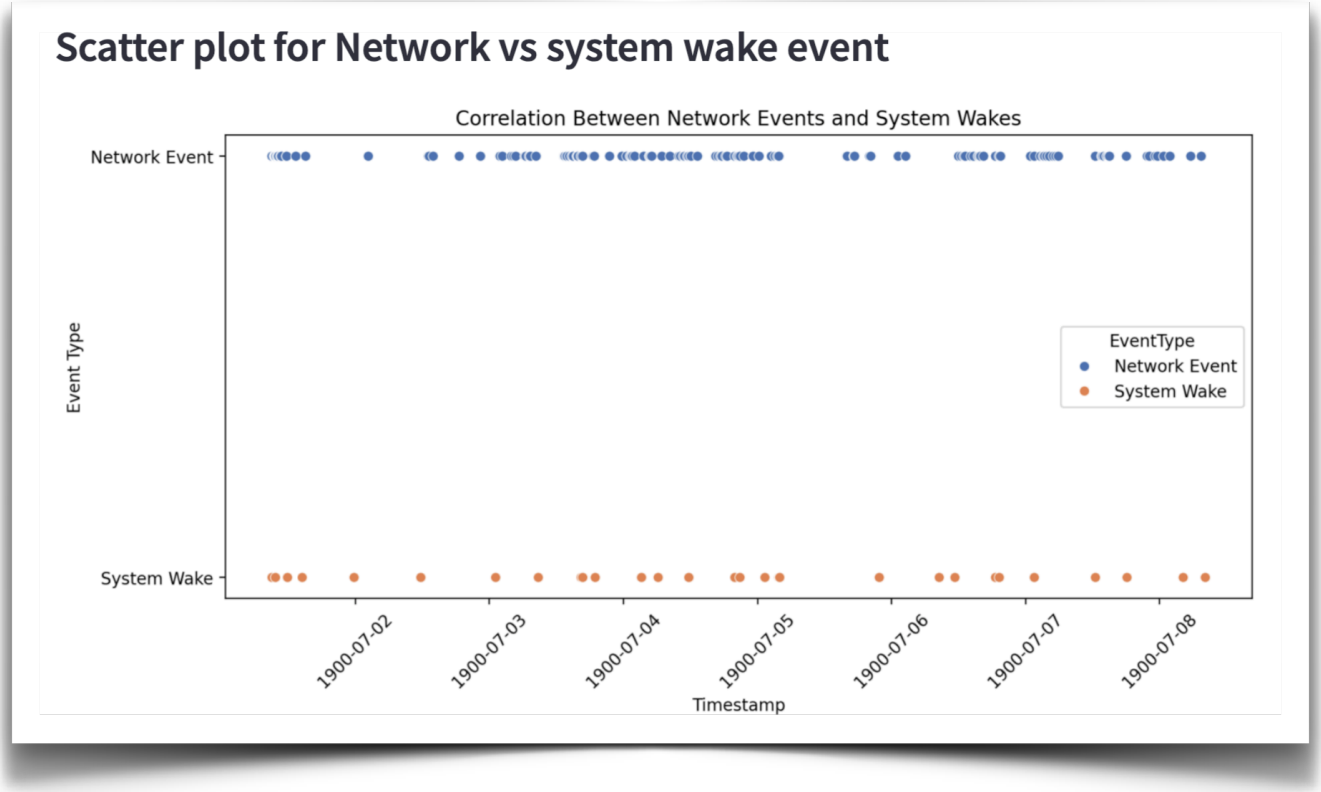


Zookeeper cluster are overloaded you should monitor specific metrics and log data that indicate the health and workload of each component/worker for example: Recvworker, Overloaded workers can show an

unusually high number of connection requests (“which might lead to out of queue issue”) and fail. Increased Latency with frequent Connection Timeouts.

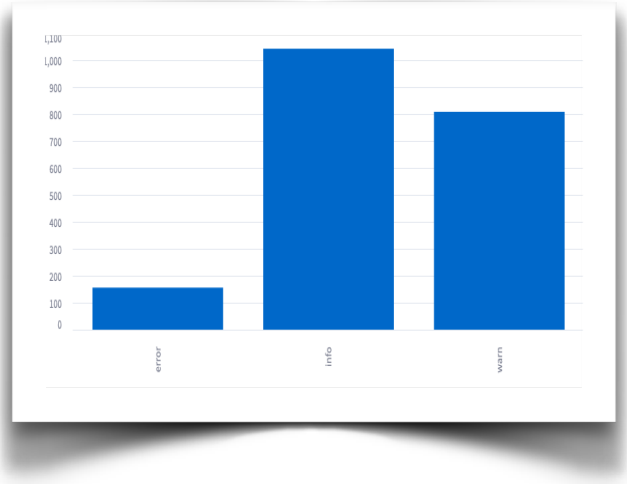
4. Can We Identify Problematic Network Events That Trigger System Wakes in Mac Logs?

Solution : Yes, problematic logs in Mac can be identified by using scatter plots and heatmaps to find network events that frequently precede system wakes, and by reviewing logs for specific high-frequency dates.

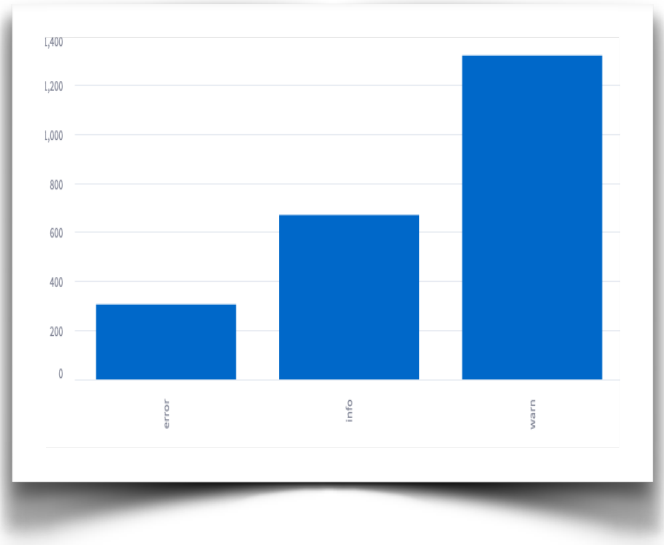


Task 5 :

1. **INFO Log Bias**: In many systems, logs are filled with an excessive number of `INFO` messages. While these messages are helpful for understanding the normal operation of a system, their sheer volume can drown out more critical log levels like `ERROR` and `WARN`. This can be problematic because it makes it difficult to quickly identify real issues that require immediate attention. For example, an important `ERROR` message indicating a significant failure might get lost among hundreds of routine `INFO` messages. This log bias can significantly impact the efficiency of monitoring and troubleshooting processes.



2. **Time-Based Verbosity Bias**: In zookeeper, show a pattern where there are sudden bursts of `WARNmessages` at specific times, followed by periods of normal logging. These spikes can create a misleading narration that the system is unstable or having issues during those times. In reality, these `WARN` messages might not indicate actual problems; they could be related to normal maintenance activities or expected temporary basss. The bias comes from interpreting these logs without context, leading to unnecessary alarms and potentially misguided troubleshooting efforts.



3. **Network vs. System Wake Bias**: In Mac operating system logs, there is often a pattern where network events are logged immediately before system wake events. This temporal sequence can lead to a biased assumption that network changes are the primary cause of system wakes.

However, this is not always the case, as system wakes can be triggered by various factors such as scheduled tasks, hardware events, or user actions. The bias occurs when logs are interpreted without considering all possible causes, leading to incorrect conclusions about the relationship between network activity and system behavior. Below shown is scatter plot indicating network system wake bias.