Abstract: Real-time Log Analysis Using Hadoop and Spark

This project, **"Real-time Log Analysis Using Hadoop and Spark,"** is all about quickly understanding what's happening with computers and software by looking at their "logs"—detailed notes about everything they do. Imagine your computer or app constantly writing a diary. There are so many entries that no human could read them all fast enough to find problems. This project uses special computer tools to read these "diaries" in real time, finding issues and telling us right away.

What Problem Are We Solving?

Computers create tons of "log" data, like a diary of everything they do. If something goes wrong, like an app crashing or someone trying to hack in, it's all written in these logs. The problems we're fixing are:

- **Too Much Data:** There's simply too much log data for people or older systems to handle.
- **Need It Now:** When a problem happens, you need to know *immediately*, not hours later, to fix it quickly.
- **Mixed Information:** Logs come in many different formats, making them hard to understand together.
- **Growing Fast:** The amount of data keeps growing, so our solution needs to grow with it.

This project builds a system that can quickly read all these logs, find out what's important, and alert us immediately. This helps us fix problems faster, keep things running smoothly, and even prevent security issues.

What Tools Are We Using?

We're using powerful tools designed for huge amounts of data:

- **Hadoop HDFS:** This is like a giant, super-safe storage locker for *all* the log notes. It keeps them organized and accessible, even if there are billions of them.
- **Apache Spark:** This is the "brain" that does the heavy lifting. It has two main parts:
  - **Spark Streaming:** This part reads the log notes *as they arrive* in real time, looking for immediate issues.
  - **Spark SQL & MLlib:** These parts help us ask smart questions about the logs and use

machine learning to find unusual patterns automatically.

- **Kaa/Flume:** These are like special couriers that pick up log notes from where they're created and deliver them to our system.
- **Python/Scala:** These are the computer languages we use to write all the instructions for these tools.
- **Grafana/Kibana:** These tools create easy-to-understand dashboards and charts that show us what's happening with our logs, so we can see trends and problems visually.

How Does It Work (The Steps)?

The project is broken down into a few main parts, like a factory line for logs:

1. **Collecting Logs:** First, we gather all the log notes from wherever they are created (computers, apps) using **Kaa/Flume**.
2. **Storing Logs:** All these collected logs are sent to our giant storage locker, **HDFS**, where they are kept safe. Spark can access them from here.
3. **Processing Logs:** This is where **Spark** does its magic. It reads the incoming logs, cleans them up, and then uses smart computer programs (**MLlib**) to find anything unusual or problematic, like a sudden increase in errors.
4. **Analyzing Logs:** We then dig deeper, extracting specific details from the log messages and calculating important numbers, like how many requests a website is getting per second.
5. **Sending Alerts:** If Spark finds something critical or unusual, it immediately sends out **alerts** (like an email or Slack message) to let people know there's a problem.
6. **Showing What's Happening:** Finally, we use tools like **Grafana/Kibana** to create real-time dashboards. These dashboards show us easy-to-understand charts and graphs of what's going on with our systems, trends, and any errors.

The whole process is a continuous loop: logs are created, collected, processed, analyzed, and then alerts or visuals are provided.

What Will This Project Achieve?

This project aims to make computer systems work better and safer. We expect it to:

- **Make Systems More Reliable:** By finding problems fast, we can fix them before they cause big issues, meaning less downtime for websites and apps.
- **Fix Problems Faster:** When something does break, we can find out why much more

quickly because we have all the log information right there.

- **Be Proactive:** Instead of waiting for things to break, we can see problems starting and deal with them *before* they become serious.
- **Improve Security:** Catching unusual activities or attempted hacks right away helps keep our systems much safer.
- **Handle Lots of Data:** The system is built to handle huge and growing amounts of log data, so it won't get overwhelmed.
- **Keep a History:** We'll have a complete history of all log data, which is useful for looking back at past issues, understanding trends, and meeting legal rules.

Basically, this project helps organizations truly understand what their computer systems are doing, in real-time, using smart technology. This makes everything run smoother and more securely.

What's Next for This Project?

We can make this project even better by:

- **Smarter Anomaly Detection:** Using even more advanced "deep learning" AI to find even subtler problems.
- **Cloud Ready:** Making it easy to run this system on big cloud services like Amazon Web Services (AWS) or Microsoft Azure, which makes it even more powerful and easier to manage.
- **Auto Log Tagging:** Using advanced text analysis to automatically categorize what each log message is about, making analysis even quicker.

24M11MC134
Parimi Renuka Chowdary
Aditya University