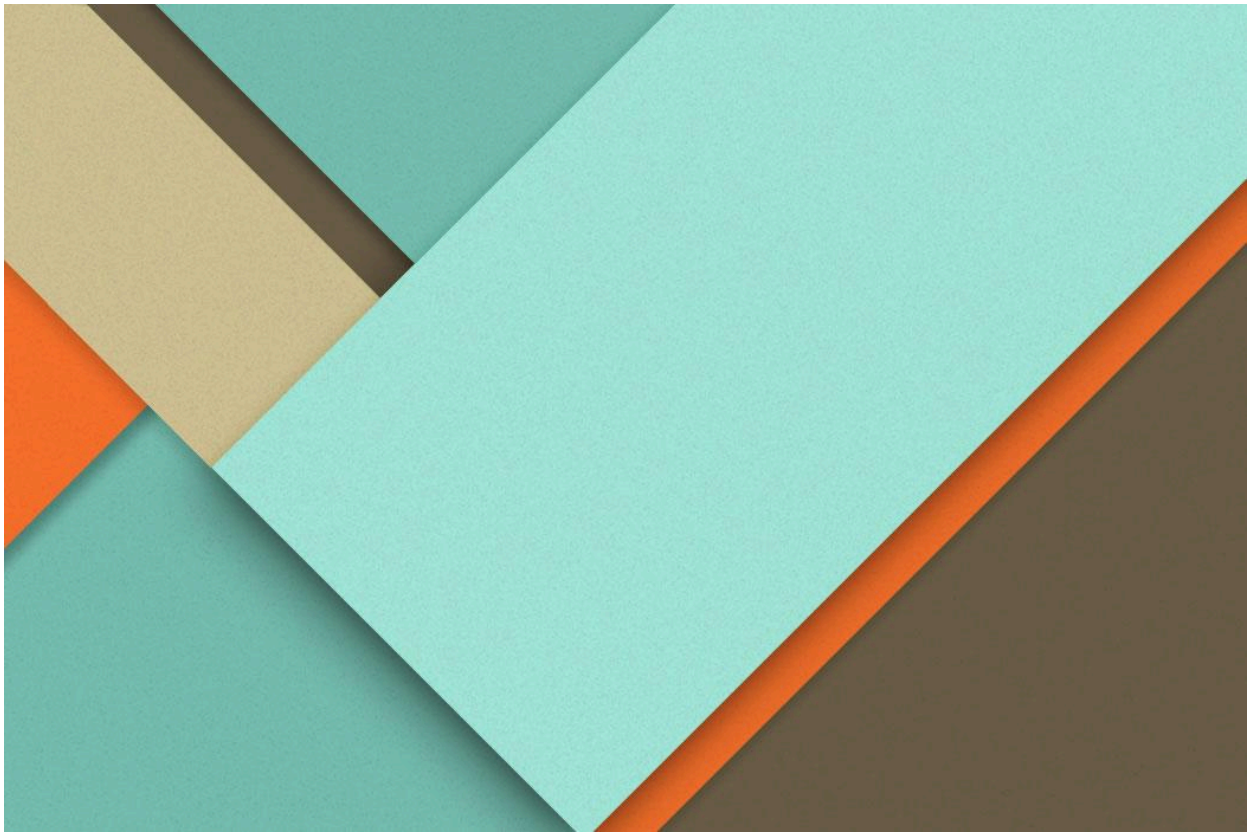

Log file analysis task

—

Nadine Rasmy
2205203

1. Introduction

This report presents the results of analyzing a web server log file using a Bash script, as per the requirements of the assignment. The objective is to extract statistical insights, identify patterns in requests, detect potential issues, and provide actionable recommendations for system improvement. The log file, assumed to be in Apache Combined Log Format, contains details such as IP addresses, timestamps, request methods (GET/POST), status codes, and more. The analysis covers request counts, failure rates, user activity, and temporal patterns, with findings summarized and interpreted to guide system optimization.

2. Methodology

A Bash script (log_analysis.sh) was developed to process the log file and compute the required statistics. The script uses standard Unix tools such as awk, grep, sort, and uniq to extract and analyze data. Key steps include:

- Counting total, GET, and POST requests.
- Identifying unique IP addresses and their request breakdown.
- Calculating failure rates (4xx/5xx status codes) and their distribution.
- Determining the most active IP and request patterns by hour and day.

The script outputs results to the console and a text file (log_analysis_report.txt), which forms the basis of this report.

3. Results

The following subsections present the statistical findings from the log file analysis.

3.1. Failure Requests

- Failed Requests (4xx/5xx): 220.
- Failed Percentage: 2.00%.

3.2. Top user

- Most Active IP: 66.249.73.135 with 482 requests.

3.3. Daily Request Averages

- Number of Days: 4.
- Average Daily Requests: 2,500.

3.4. Failure Analysis

- The days with the highest number of failed requests

19/5/2015 → 66

18/5/2015 → 66

20/5/2015 → 58

17/5/2015 → 30

3.5. Requests by Hour

- The number of requests per hour, highlighting peak usage times. (full data is in the script)

Hour 8 → 345

Hour 22 → 346

Hour 3 → 354

Hour 14 → 498

Hour 15 → 496

Hour 19 → 493

3.6. Status Codes Breakdown

- The frequency of HTTP status codes.

200 → 9126 requests.

304 → 445 requests.

404 → 213 requests.

301 → 164 requests.

206 → 45 requests.

500 → 3 requests.

416 → 2 requests.

403 → 2 requests.

3.7. Most Active User by Method

- Top GET User: 66.249.73.135 with 482 GET requests.
- Top POST User: 78.173.140.106 with 3 POST requests.

3.8. Patterns in Failure Requests

20/5/2015:09 → 15 failures.

20/5/2015:05 → 9 failures.

19/5/2015:06 → 9 failures.

19/5/2015:01 → 8 failures.

17/5/2015:17 → 7 failures.

4. Analysis

The analysis reveals several key patterns and insights:

- Peak Usage Times: Request volume peaks between 14:00 and 20:00, with the highest at 14:00 (498 requests). This indicates high user activity during these hours, likely corresponding to business or peak browsing times.
- Failure Patterns: The majority of failures (220, or 2.00%) are 404 errors (213), suggesting broken links or missing resources. Failures are concentrated on 19 and 18 May 2015 (66 each), with a notable spike at 09:00 on 20 May 2015 (15 failures), possibly due to server overload or configuration issues.
- User Activity: IP 66.249.73.135 dominates with 482 GET requests, which may indicate a web crawler. POST requests are minimal (e.g, 3 from 78.173.140.106), suggesting limited interactive usage.
- Status Codes: Most requests are successful (9,126 for 200, 445 for 304), but 404 errors (213) and rare 500 errors (3) warrant attention.

5. Suggestions

Based on the findings, the following recommendations are proposed to improve system performance and reliability:

1. Address 404 Errors: Review and fix broken links or missing resources causing 213 404 errors to enhance user experience.

2. Investigate Server Errors: Examine server logs for 19 and 20 May 2015, particularly at 09:00 on 20 May (15 failures), to identify causes of 500 errors (3) and high failure rates.

3. Optimize for Peak Hours: Increase server resources (e.g., CPU, memory) between 14:00 and 20:00 to handle peak loads (475–498 requests).

4. Monitor High-Activity IPs: Verify if IP 66.249.73.135 (482 GET requests) is a legitimate crawler using tools like whois. If malicious, implement rate-limiting to prevent abuse.

5. Implement Real-Time Monitoring: Deploy tools to detect and alert on 4xx/5xx errors in real time, enabling faster response to issues.

6. Conclusion

The log file analysis successfully extracted key statistics and patterns, revealing peak usage times, failure concentrations, and dominant user activity. The system performs well overall, with a low failure rate (2.00%), but specific issues like 404 errors and occasional server errors require attention. The proposed suggestions aim to enhance reliability, optimize performance during peak hours, and address potential security concerns. Further analysis with additional log data could provide deeper insights.