Name: Arsany Osama ID: 2205122

Apache Log Analysis Report

Log File: apache_logs

Summary of Findings

This report analyzes the Apache log file *apache_logs*, covering a period of approximately 4405 days, from 17 May 2015 to at least 20 May 2015. The analysis provides key insights into server performance, request patterns, user behavior, and potential issues such as failures and security concerns.

Output from the analysis of the bash script file

- **Total Requests:** 10,000
- **GET Requests:** 9,952 (99.52%)
- **POST Requests:** 5 (0.05%)
- **Unique IP Addresses:** 1,753
- **Failed Requests (4xx/5xx):** 220 (2.00%)
- **Most Active IP:** 66.249.73.135 (482 GET requests), 78.173.140.106 (3 POST requests)
- **Average Requests per Day:** 2.27 over 4405 days
- **Peak Request Hour:** Hour 14 (498 requests)
- **Highest Failure Hour:** Hour 09 (18 failed requests)

Detailed Analysis

1. Request Counts

The majority of traffic consists of GET requests (9,952), indicating resource retrieval dominates server usage. POST requests are minimal (5 total), while 3 HEAD requests were also observed, likely from clients checking resource metadata.

2. Unique IP Addresses

There were 1,753 unique IP addresses recorded, reflecting a diverse user base accessing the server over the analyzed period.

3. Failure Requests

Failed requests numbered 220, about 2% of total requests. Breakdown of status codes shows:

- **404 Not Found:** 213 occurrences
- **500 Internal Server Error:** 3 occurrences
- **416 Range Not Satisfiable:** 2 occurrences
- **403 Forbidden:** 2 occurrences

The dominance of 404 errors suggests broken links or missing resources are a major issue.

4. Top Users

The most active IP, (66.249.73.135), made 482 GET requests and is likely a web crawler such as Googlebot. For POST requests, IP (78.173.140.106) issued 3 of the 5 total POST requests, indicating unusual activity warranting monitoring.

5. Daily Request Averages

Averaged across 4405 days, there were 2.27 requests per day, indicating low traffic which might reflect incomplete logs or sparse activity over a long timeframe. For example, on 17 May 2015, daily requests ranged from 1 to 5 per timestamp with no clear high-activity pattern.

6. Days with Highest Failures

Days with multiple failed requests include 20 May 2015, which had the highest number of failures at various timestamps, followed by 17, 18, and 19 May 2015 showing some failed requests as well.

7. Requests by Hour

Request volume peaked in the early afternoon at Hour 14 with 498 requests and was lowest during Hour 08 with 345 requests. Failure counts were highest in the morning hours, particularly Hour 09, indicating possible issues during that period.

8. Status Code Breakdown

- **200 OK:** 9,126 requests (91.26%)
- **304 Not Modified:** 445 requests (4.45%)
- **301 Moved Permanently:** 164 requests (1.64%)
- **404 Not Found:** 213 requests (2.13%)
- **Other errors like (500, 416, 403):** few occurrences (<0.1% each)

9. Failure Patterns

Failures were concentrated in morning hours; 18 failures occurred at Hour 09, followed by 15 and 14 at Hours 05 and 06. The lowest failures were during Hour 08 (2 failures). This pattern suggests possible issues related to automated tasks or maintenance in early hours.

10. Request Trends

Afternoon hours show peak activity while mornings show higher failure rates. The overall low average daily requests hint at either incomplete data or a low-traffic server.

Suggestions Based on Results

1. Reducing Failures

- **404 Errors:** Audit website for broken links, implement redirects for outdated URLs, and maintain a sitemap or CMS (Content management system) to ensure resource availability.
- **500 Errors:** Review server logs around error times and improve error handling to reduce internal server errors.
- **416 and 403 Errors:** Verify server handling of range requests and review access controls for forbidden errors.

2. Addressing Critical Times based on the request patterns and failure trends

- Investigate morning failure spikes around Hours 05, 06, and 09 for possible maintenance or automated scripts causing issues.
- Ensure server capacity and optimization for peak traffic around Hours 14 (2 PM) – 15 (3 PM), possibly using caching or load balancing. [Hour 14: 498 requests, Hour 15: 496 requests]
- I suggest that we have to review logs from high-failure times such as 20 May 2015 in Hour 09 (18 failed requests) to determine root causes.

3. Security Concerns

- We have to check legitimacy of high-volume for IP (66.249.73.135) by verifying user-agent strings.
- Monitor and validate POST requests from IP (78.173.140.106) given their rarity and potential for malicious activity.
- I suggest to consider implementing rate-limiting, Web Application Firewalls (WAF), and review unusual request patterns regularly.

4. My Suggestions for System Improvement based on analysis

- Implement caching solutions to reduce load during peak hours.
- Optimize back-end processes to reduce [500 errors] especially during failure peaks.
- Set up real-time monitoring and alerting for spikes in failed requests using modern tools like Prometheus (open-source systems monitoring and alerting toolkit)
- Improve user experience by customizing 404 pages and managing redirects effectively.
- Ensure full retention of log data to improve accuracy of long-term analysis and trend detection.

Conclusion

The Apache log analysis indicates an overall reliable server with a low failure rate of 2%, largely affected by 404 errors due to missing resources. Morning hours exhibit higher failures, requiring investigation and possible maintenance changes. Peak afternoon traffic suggests focus on performance optimization. Monitoring and addressing the unusual activity from specific IPs will bolster security. Implementing the outlined suggestions can enhance server reliability, user satisfaction, and security posture.