CMPSC 311

Project and Internship Experience Report

Nandan Hirpara

Introduction

The internship experience has been an enriching journey in the realm of systems programming and software development. Engaged in a challenging project, I had the opportunity to contribute to the creation of a sophisticated software-based RAID (Redundant Array of Independent Disks) system tailored for JBOD (Just a Bunch of Disks) storage configurations. This report delves into the detailed analysis of the project, emphasizing the codebase's intricacies and my individual contributions.

Project Overview

Project Description

The overarching goal of the project was to design and implement a robust software RAID system capable of efficiently managing JBOD disks. The project scope encompassed critical functionalities, including JBOD disk mounting and unmounting, optimized read and write operations, a dynamic caching system, and seamless communication with the JBOD server.

Project Objectives

1. **JBOD Disk Management:** Develop functions for mounting and unmounting JBOD disks, ensuring proper system initialization and shutdown procedures.
2. **Read and Write Operations:** Implement algorithms for reading and writing to JBOD disks, optimizing data access and transfer efficiency.

3. **Caching Mechanisms:** Integrate a dynamic caching system to enhance read and write performance by reducing the frequency of disk access.

4. **Networking:** Establish reliable communication between the software RAID system and the JBOD server to execute JBOD operations seamlessly.

Detailed Code Analysis

mdadm.c

Functions:

- **Mounting and Unmounting Disks (mdadm_mount and mdadm_unmount):** Responsible for managing the mount status of JBOD disks, preventing redundant mounting or unmounting.

- **Read and Write Permissions (mdadm_write_permission and mdadm_revoke_write_permission):** Handle the granting and revocation of write permissions to JBOD disks.

- **Read Operation (mdadm_read):** Ensures the validation of the mount status before performing efficient read operations from JBOD disks.

- **Write Operation (mdadm_write):** Manages the writing of data to JBOD disks, incorporating proper initialization and handling of caching mechanisms.

Key Features:

- **Efficient Disk Access:** Functions are designed to optimize disk access, minimizing unnecessary operations and improving overall system performance.
- **Permission Handling:** Provides proper control overwrite permissions, ensuring data integrity and security.

cache.c

Functions:

- **Cache Management (cache_create and cache_destroy):** Responsible for creating and destroying the cache, ensuring efficient memory utilization.
- **Cache Lookup (cache_lookup):** Checks if a specific block is present in the cache, minimizing disk accesses for frequently accessed data.
- **Cache Update (cache_update):** Updates the content of a cache entry when the corresponding block is modified.
- **Cache Insertion (cache_insert):** Inserts a new entry into the cache, optimizing write operations by reducing disk accesses.

Key Features:

- **Dynamic Cache Management:** The cache can be resized dynamically, adapting to changing workloads and storage requirements.
- **Clock Algorithm:** The clock algorithm is employed for cache replacement, efficiently managing cache entries based on usage patterns.

net.c

Functions:

- **Connecting and Disconnecting (jbod_connect and jbod_disconnect):** Manages the establishment and termination of connections with the JBOD server.
- **Client-Server Communication (jbod_client_operation):** Facilitates communication with the JBOD server, executing JBOD operations and handling server responses.

Key Features:

- **Network Protocol:** Adheres to a well-defined network protocol, enabling seamless communication between the RAID system and the JBOD server.
- **Error Handling:** Includes error-checking mechanisms to deal with potential communication issues, ensuring the reliability of network operations.

Internship Experience

Learning Opportunities

The internship provided an immersive learning experience in system-level programming, RAID systems, and network communication. The exposure to real-world applications of theoretical knowledge enhanced my understanding of storage systems.

Challenges Faced

Working on a complex project like the software RAID system presented challenges, particularly in debugging disk operations, implementing caching mechanisms, and ensuring efficient network communication. Overcoming these challenges improved my problem-solving skills and ability to navigate intricate codebases.

Individual Contribution

My primary contribution involved the development and optimization of the caching mechanism. Implementing dynamic cache management strategies and the clock algorithm was a critical aspect of improving the overall performance of the RAID system.

Future Enhancements

1. **Performance Optimization:** Further optimize disk read and write operations to enhance overall system performance.
2. **Redundancy and Fault Tolerance:** Implement RAID levels with redundancy and fault tolerance mechanisms for improved data integrity.
3. **Dynamic Cache Management:** Develop adaptive cache management strategies to optimize cache hit rates based on varying workloads.
4. **Security Measures:** Integrate robust security measures to ensure data privacy during transmission and storage.

5. **Scalability:** Enhance the system's scalability to accommodate a larger number of disks and adapt to evolving storage requirements.
6. **User Interface:** Consider developing a user-friendly interface or a command-line tool to facilitate easier system management.

Conclusion

The internship experience in developing a software-based RAID system for JBOD configurations has been both challenging and rewarding. The project has provided a deep dive into system-level programming, networking, and storage management. The skills acquired and challenges overcome during this internship have significantly contributed to my growth as a software developer.

The detailed code analysis presented in this report showcases the intricate design and functionality of the software RAID system. Future enhancements outlined indicate a clear path for further refinement and expansion of the system's capabilities.

In conclusion, the internship has been instrumental in expanding my technical skill set, fostering teamwork, and preparing me for future endeavors in the field of systems programming and software development.