Student internship report

Internship Unit: Institute of Parallel and Distributed Systems (IPADS)

Internship Period: 24/06/2024 to 04/08/2024

College: School of Electronic Information and Electrical Engineering

Major: Software Engineering

Student Name: *VAHAGN* Student Number: 521030990006

VAHAGN

2024 - 09 - 14

The main contents of the internship report include the purpose and tasks of the internship, the internship unit, the internship content, the internship gains, etc.

1. Introduction

From June 24, 2024, to August 4, 2024, I completed an internship at the Institute of Parallel and Distributed Systems (IPADS) at Shanghai Jiao Tong University (SJTU). I worked with a group of four classmates under the supervision of Teacher Du on a project focused on improving Linux kernel memory resource management, specifically optimizing the use of the madvise() system call. This experience helped deepen my understanding of operating systems and provided valuable insight into real-world computer system research processes and collaboration.

2. Internship Purpose

The purpose of this internship was to analyze and improve Linux memory resource management using the madvise() system call. The project focused on addressing the limitations of madvise() in real-world applications and optimizing its performance by suggesting a new interface called smart-madvise, which is a dynamic and global memory management optimization system. Smart-madvise is a kernel module which collects real-time memory usage data, allowing the kernel to make more informed decisions for memory optimization across multiple applications, rather than relying on static hints provided by individual processes.

3. Tasks

1. Analyzing madvise() Code and Running Performance Tests: I thoroughly studied the source code of madvise() to understand its inner workings and limitations.
2. Application Benchmarking: I identified and analyzed madvise() usage in real-world applications, such as bbolt, and evaluated various operation types and metrics.
3. Designing Smart-Madvise: I worked on the smart-madvise module, focusing on developing the **data memory usage information collector** to provide real-time memory insights for kernel optimization decisions.

4. Content

During the internship, I contributed to the development of the smart-madvise system, which is designed to optimize memory management in Linux by dynamically adjusting memory strategies based on real-time usage patterns. Before diving into the smart-madvise implementation, I first analyzed how the existing madvise() system call functioned in real-world applications.

4.1 Analyzing madvise() in bbolt

bbolt is a fast, embedded key-value database that operates in memory. It was chosen as a testbed to understand the limitations of madvise(). I read the bbolt source code and ran benchmarks on it to observe how madvise() was used and how it impacted memory performance. Through this, I identified several limitations of madvise():

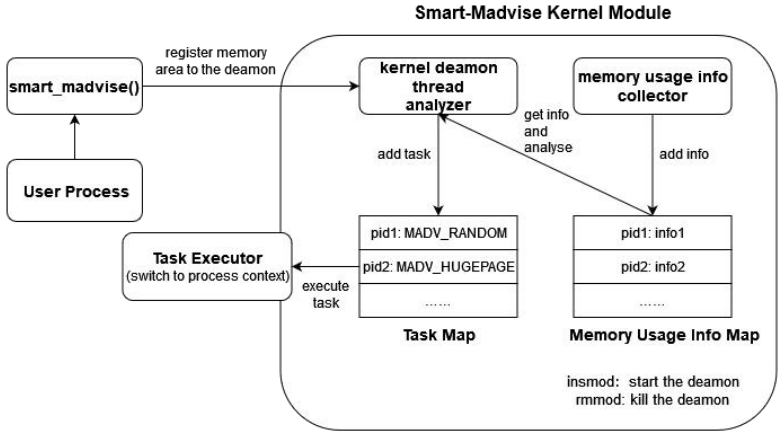
1. It provides static memory hints, which cannot adapt to changing memory usage patterns in real time.
2. The performance benefits of madvise() were limited in scenarios where memory access patterns changed dynamically.

This understanding informed the design and implementation of smart-madvise, which aimed to address these issues by dynamically adjusting memory strategies based on real-time data.

4.2 Overview of Smart-Madvise Implementation

Smart-madvise operates by continuously collecting and analyzing memory usage data to dynamically adjust memory management strategies. The system consists of key components as shown in the diagram:

- Memory Usage Info Collector: Gathers real-time memory usage data from various processes.
- Kernel Daemon Thread Analyzer: Analyzes the collected data and adds tasks to the Task Map, determining whether the process should switch to memory strategies like MADV_RANDOM or MADV_SEQUENTIAL.
- Task Executor: Executes the memory optimization tasks by applying the appropriate madvise() call to the relevant processes.



4.3 My Role: Developing the Memory Usage Information Collector

I was responsible for implementing the memory usage info collector. This component tracks real-time memory access patterns for each process, specifically determining whether the access pattern is sequential or random.

My work involved:

1. Tracking Access Patterns: I designed a mechanism that monitors page faults and compares memory addresses to determine if they are being accessed **sequentially or randomly.** If sequential, the sequential access counter is incremented; if random, the random access counter is updated.
2. Data Structures for Process Tracking: Since processes could be interrupted or terminated, I helped to develop specialized data structures to track the memory usage patterns for each process safely. This ensured that memory tracking would remain robust even when processes were killed or interrupted unexpectedly.
3. Dynamic Strategy Switching: Based on the collected data, the system dynamically switched between memory optimization strategies. For example, if a process exhibited mostly sequential memory access, it would be flagged to use MADV_SEQUENTIAL. If the pattern became more random, it would switch to MADV_RANDOM.

This dynamic adaptation was key to the effectiveness of smart-madvise, as it allowed for more efficient memory management based on the real-time behavior of running applications.

3. Internship Gains

Through this internship, I:

1. Enhanced Technical Skills: Improved my C programming, enhanced my Linux system understanding and gained experience in kernel module development.
2. Deepened OS Knowledge: Gained a better understanding of Linux memory management and the potential of madvise().
3. Learned Teamwork: Realized the importance of being a team player over trying to do everything alone. Efficiently cooperated with team members through division of labor and regular communication.
4. Prepared for Future Research: The experience motivated me to continue research in computer systems, understanding real research dynamics, challenges, and teamwork.
5. Received Valuable Guidance: Regular meetings with our supervisor, Teacher Du, provided direction, feedback, and answers to my questions.

4.4 Future Plans

In the future, I plan to integrate my page access pattern detection code into BCC (BPF Compiler Collection), a toolkit for creating efficient kernel tracing tools. Additionally, I have developed a strong interest in kernel module programming, and I am now actively reading more Linux blogs to deepen my knowledge. I aim to contribute to the Linux kernel community in the near future.

指导教师对学生实习情况的评价意见	vahagn 的实习内容是参与 SmartMadvise 项目的开发，主要聚焦在探索更优的内存策略、对当前系统使用 madvise 的表现进行测试评估、内核扩展、以及最终的系统测试等部分。Vahagn 在整个过程中表现优异，作为留学生，能够积极主动和小组其他同学进行交流，并且开发能力、调研能力等多方面都较为优秀。整体来看，Vahagn 同学的实习表现十分优异。 Vahagn's internship involved participating in the development of the SmartMadvise project, focusing on exploring better memory strategies, testing and evaluating the performance of the current system using madvise, kernel expansion, and final system testing. Vahagn performed well throughout the entire process. As an international student, he was able to actively communicate with other students in the group, and his development and research capabilities were relatively good. Overall, Vahagn's internship performance was excellent. *Instructor (Signature):* 指导教师（签名）：  2024 - 09 - 16
