Visualization Toolkit for Linux Kernel Tracing and Performance Verification

Jaeman Yoo, Yonghwan Jeon, Minsu Kang, Dongyeon Kim, Junghoon Shin, Sangjun Lee
School of Computing, Soongsil Univ.
Seoul, Korea
{woaksdosl10, dlsrp12369}@gmail.com, {roroeiro, dongyeon89, junghoon.shin, sangjun}@ssu.ac.kr

Abstract
Linux is one of the most representative operating systems. And Linux kernel is currently being used in various fields and developed by many researchers and developers. Linux kernel developers need a toolkit to monitor the performance of developed systems and detect errors in order to meet the expectations. In this paper, we propose Kernel-Ray that is the visualization toolkit for Linux. It is able to trace the internal functions of Linux kernel and visualize the internal operations of the system using graphs and tables. Also it expresses the interruption latency of Linux kernel in graphs, and thus easily allows Linux kernel developers to monitor performance and errors.

Key Words
Design and Development, Verification and Validation
Linux kernel, Tracing and Monitoring

1. Introduction
As the IT field has developed and the perceptions of software have been raised in recent years, the importance of software as well as hardware has been highlighted. Accordingly, demands for software quality and performance optimization have been also on the rise. In addition to scientific interests, a better understanding of motivational processes within the Linux kernel project might help to improve software development processes in other open source projects as well as in the corporate world[1].

The Linux kernel have been used in various fields from the server system up to the embedded system. Therefore, system software developers who wish to become involved in the development and maintenance of Linux are increased. Some of these developers are motivated by personal interest, some work for Linux companies, some work for hardware manufactures, and some are involved with in-house development projects[2][3].

Linux kernel developers use toolkits to measure the kernel performance to test the performance of a developed system and analyze performance with a toolkit to trace kernel operations in order to improve the latency and throughput of system or software and use it more efficiently[4]. The existing kernel tracing method can analyze kernel tracing information using proc file systems and event tracers, but it has a couple of problems including not being able to trace functions active inside the kernel and having a difficult time with analysis tracing results due to providing the tracing result data in a form of lower legibility by simply enumerating the data. In addition, existing visualization tools for recording a tracking kernel operations have a disadvantage in that user access is difficult by using command interface.

In an effort to overcome those problems, this paper presents a description of design and implementation of a toolkit that allows developers to perform the functions of tracing kernel operations and measuring kernel performance easily through the graphic user interface and visualizes the results in graphs and tables for users to easily understand.

2. Related Work

2.1 Ftrace

Ftrace is the tracer framework of the Linux kernel that provides the standard tracing the internal operations of Linux kernel[5][6][7]. Ftrace supports two major functions: one is tracing based on the plugin tracer, and the other is the trace event to trace evens inside the kernel. The plugin tracer helps with kernel tuning and development. There are supports for many different types of plugin tracers to trace process schedules and interruption latency and profile function calls[4][8]. The event tracer[4] used to be based on tracer pointers in the past and records the processing of representative processing points inside the kernel[4][10]. The remainder of this paper is organized as follows. Section 2 explains related work. Section 3 describes the proposed system. Finally, Section 4 presents the conclusion.

2.2 Cyclictest

Cyclictest[9] is a tool to evaluate performance real-time during rt-tests. Cyclictest measures kernel performance by generating threads whose timers expire at a certain interval and measuring time from the occurrence of interruption following the expiration of a timer to the preoccupancy of CPU by the threads.
Users can assess kernel performance by setting an array of options related to the threads including the expiration time of a timer, the number of threads, and the priority of threads. The reliability in the measured latency values depends on the timer used for measurement.

3. Proposed System

In this paper, we have developed Kernel-Ray that is a visualization toolkit for Linux kernel tracing and performance verification. Our proposed system was designed to facilitate debugging and error monitoring to developers of Linux kernel. Figure 1 shows overall architecture of the Kernel-Ray. Kernel-Ray consists of the Trace View and Latency View. Each part is responsible for the trace and performance verification of Linux kernel.

3.1 Trace View

Trace View provides the trace function of kernel operations and visualization of trace result data using the graphs and tables for users. Trace View consists of the following modules.

- **Ftrace GUI (Graphical User Interface)**: Ftrace GUI is able to configure the tracing of kernel operations using plug-in tracer by users.
- **Event Tracer GUI**: Event Tracer GUI provides to configure the event tracing of kernel using the event tracer.
- **Trace Data Parser**: Trace Data Parser functions to load and parse data recorded in the kernel after operation of the plug tracer or event tracer that is selected by user.
- **Trace Data Viewer**: Trace Data Viewer shows the parsed data by Trace Data Parser using the table view and graph view.

3.2 Latency View

Latency View provides the real-time verification of Linux kernel performance by measuring the latency time of the kernel and graphs about latency time data. Latency View consists of following modules.

- **Cyclictest GUI**: Cyclictest GUI serves users to configure a kernel module that provides the performance verifications of Linux kernel using the Cyclictest.
- **Latency Data Parser**: Latency Data Parser functions to load and parse data recorded in the kernel after operation of the Cyclictest.
- **Latency Data Viewer**: Latency Data Viewer shows the parsed data by Latency Data Parser using the graph view.

3.3 Tracing of Linux kernel Operations

The function of tracing kernel operations of Kernel-Ray proposed in the study consists of the setting of kernel operations tracing mode and the component to show resulting data. Figure 2 offers the overview of kernel operation tracing function[12]. The function of tracing kernel operations in Kernel-Ray includes Ftrace and Event Tracer, and each function has a dialogue to set its own options.
3.4 Kernel Performance Verification

The function of measuring kernel latency in Kernel-Ray mainly includes the graph view, option layout, and resulting data layout. Figure 3 shows the overview of kernel performance verification function.

4. Conclusion

In this paper, we proposed a method to design and implement Kernel-Ray capable of assessing the performance and function of kernel operations. Kernel-Ray is a kernel debugging and verification program applicable to all systems based on Linux and RTLinux and expected to help Linux system developers increase software quality and cut down product development expenses through performance monitoring and error detection by being applied to the field of system software and middleware development.

Acknowledgement(s)

This research was supported by the MSIP (Ministry of Science, ICT and Future Planning), Korea, under the Seoul Accord Vitalization Program (IITP-2015-R0613-15-1175) supervised by the IITP (Institute for Information & communications Technology Promotion) and supported by the MSIP (Ministry of Science, ICT & Future Planning), Korea in the ICT R&D Program 2013 (No. 13-912-03-003).

References