

OS noise Analysis on Azalea-unikernel

Seung-Jun Cha, Seung Hyub Jeon, Yeon Jeong Jeong, Jin Mee Kim, Sungin Jung

*Electronics and Telecommunications Research Institute (ETRI)
Daejeon, South Korea*

{seungjunn, shjeon00, yjjeong, jinmee, sijung}@etri.re.kr

Abstract— Operating System noise is a well-known problem that occurs mainly in large-scale systems running HPC application, and it also limits performance scalability. In order to reduce OS noise occurring in a general-purpose OS, different design approached kernels like micro-kernel and lightweight kernel are studied. Recently unikernel optimized for application program has been studied. The purpose of this paper is to measure OS noise in unikernel. We compared OS noise of Linux, general-purposed OS, and Azalea-unikernel, one of the unikernels. Well-known OS noise measurement benchmarks such as fixed work quanta (FTQ), fixed time quanta (FWQ), and Hourglass were used. As a result of the experiment, it was confirmed that Azalea-unikernel has less OS noise than Linux, and it also means unikernel is a kernel suitable for supporting scalability on manycore systems.

Keywords— operating systems, OS noise, interference, Azalea-unikernel

I. INTRODUCTION

OS noise is a well-known problem in high-performance computing (HPC) environments. The HPC application performs an operation in the calculation phase and collects and processes the result in the rendezvous phase [1]. The problem of performance degradation occurs in the rendezvous phase. The investigation found that if one process is delayed by OS interference such as timer interrupt, the other processes wait while wasting their computational power in the rendezvous phase, resulting in a delay in the entire system [2, 3]. This problem affects performance even more as the number of cores increases, which leads scalability problems [4, 5].

Lightweight kernel (LWK) and micro-kernel are different approached to reduce OS noise in HPC systems. Many supercomputers use the LWK of Compute Node Kernel (CNK) [6] or Catamount [7]. In the past, in applications that mainly used scientific operations and MPI implementation, the LWK showed good performance.

The OS research is gradually evolving from the existing general-purpose OS to a specialized OS to further reduce interference and improve application performance. The HPC community is trying to apply unikernel in particular [8]. Unikernel is a specialized kernel that uses a single address space based on library OS. This has strengths in small size, performance, predictability, and security. Also, it is suitable for a virtualization environment because the execution environment works well on the hypervisor. Unikernel is

effective in terms of OS noise because the kernel and runtime library consists of only the minimum necessary functions.

In this paper, we compare the OS noise of Azalea-unikernel with Linux, a general-purpose operating system. Azalea-unikernel is one of the unikernels, which is lighter than LWK and is compatible with existing Linux binaries [6]. We first describe the definition of OS noise and measurement methods. Fixed work quanta (FTQ), fixed time quanta (FWQ), and Hourglass are well-known micro benchmarks that measure OS noise. In the testbed, OS noise was measured and compared between Linux and Azalea-unikernel. As a result, we found that Azalea-unikernel is better than Linux in terms of OS noise.

The rest of the paper is organized as follows. In Section II, we give a brief overview of Azalea-unikernel, and Section III describes OS noise and finds out about their measurement. Section IV explains the testbed and the comparison results of OS noise measurement. We conclude the paper in Section V.

II. AZALEA-UNIKERNEL

Azalea-unikernel is a library OS-based operating system and provides a system environment that can optimally execute one process [9]. Through this, a single process directly accesses the hardware and manages only the resources allocated to it. It does not cause resource competition between other processes, so the performance of the application can be improved. In addition, since only the hardware resources and kernel functions required by a specific application program are included, the attack surface is relatively small, thus reducing the potential security risk. Figure 1 shows the difference between the existing OS and Azalea-unikernel structure.

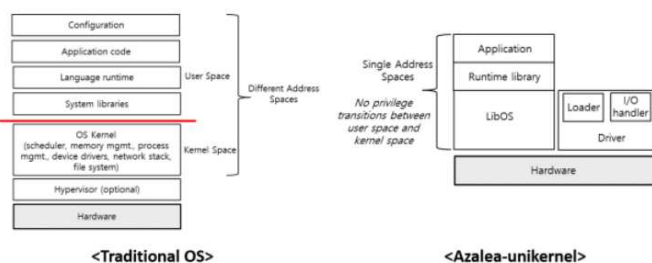


Figure 1. The structural difference between traditional OS and Azalea-unikernel

The characteristics of Azalea-unikernel are as follows: 1) Since the kernel and the application using the same address space, the overhead such as having to replace privileges is reduced, making it lighter than the LWK. 2) It provides the capability to run legacy applications. Existing source codes can be recompiled with minimal modifications or statically built Linux binaries can be executed immediately. 3) I/O performance is improved by offloading and processing I/O operations to the full-weight kernel (FWK) [10].

III. OPERATING SYSTEM NOISE

A. What is OS noise

The general definition of OS noise is the disturbance caused by the operating system when an application is running on a processor. The CPU is wasting its computational power due to such interference, and this noise can degrade the performance of parallel applications. As the number of cores increases, it affects the performance more, resulting in scalability problems.

OS noise can occur due to different kernel activities. In [11], kernel activities that cause OS noise are classified into 5 categories: periodic, page fault, scheduling, pre-emption, and I/O. OS noise includes not only the interruption of code execution caused by timer/device/software interrupts, faults, etc. but also disturbances of the memory subsystem that occur when the operating system makes requests to the processor's cache subsystem [6].

B. Measuring OS noise

A common way to measure OS noise is to perform a microbenchmark and measure the amount of computation per time interval or the time to perform the computation. Well-known benchmarks include the Fixed work quanta (FTQ) [12][13][14], fixed time quanta (FWQ) [15][16], and Hourglass [17] benchmarks.

1) FTQ / FWQ

FTQ and FWQ are effective benchmarks for measuring the interference of the operating system that receives and analyses values from running threads.

FTQ benchmark analyses operating system activity using a time-based method rather than a workload-based method. In other words, it measures the amount of workload performed during a specific time rather than fixing the workload and measuring how long it runs. It is designed to strictly adhere to the interval, in very short units, around 1ms in time.

FWQ benchmark is measured based on the time it takes to run a given workload. Since the amount of data in workload is fixed, it is effective for the statistical analysis of noise. These analysis results are divided into the amount of noise in hardware and software environments. The time-based approach provides far more detailed analysis results than the workload-based approach.

2) Hourglass

Hourglass benchmark was introduced for the analysis of the time slices assigned by the operating system to multiple

processes. In this work, it is used for checking gaps mainly generated by the operating system during execution time. The benchmark keeps reading the processor's time stamp counter. Except for some blocked system services, it is bound to a specific core and operates in the system. In other words, the wider gaps between two read operations, the more the kernel is running which means OS noise such as scheduling or interrupt handling.

IV. EXPERIMENTAL RESULTS

In this section, we explain the testbed of hardware and software for the experiments. Also, the OS noise of Linux and Azalea-unikernel was measured and analyzed for comparison on the testbed.

A. Testbed setup

All experiments are carried out on an Intel 56-core NUMA machine with 192G memory. There are two Intel Xeon processors in the system and each chip accommodates 28 cores. For the experiments, hyperthreading, turbo mode, and p- c-state were disabled. We use CentOS Linux with kernel version 2.6.14 as the operating system. Most experimental results come from the clean kernel without debugging and statistics capabilities.

B. OS noise comparison

TABLE 1. RESULTS OF THE HOURGLASS BENCHMARK

OS	Gaps	
	Avg	Max
Linux	54.37	148
Azalea-unikernel	54.38	128

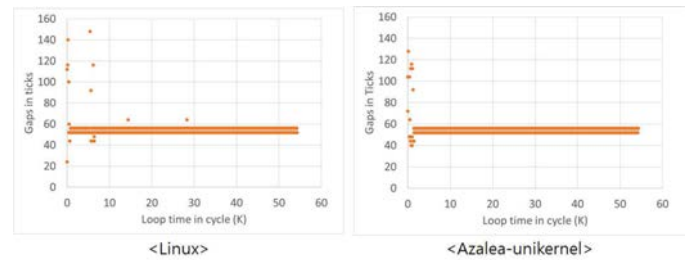


Figure 2. OS noise comparison result using hourglass benchmark

Table 1 shows the maximum and average gaps of the CPU cycle in Linux and Azalea-unikernel, while Figure 2 shows a histogram for intervals over 1000 cycles. The average number of the cycle for the timestamp count is constant at about 54. The maximum value is more than 20 for Azalea-unikernel than Linux. Also, in Linux, the performance is more distributed and thus the latency is high. This is due to noise from the operating system such as interrupts or other tasks running at the same time.

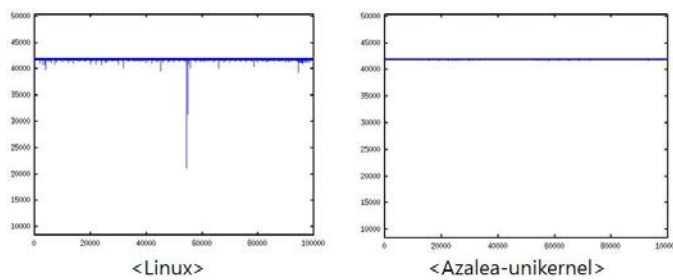


Figure 3. OS noise comparison result using FTQ benchmark

Figure 3 is the result of executing FTQ on Linux and Azalea-unikernel. The X-axis represents the sample number, and the Y-axis represents the number of iterations performed during the time quantum. Linux shows large amounts of high-frequency noise and occasional large spikes. However, we found the constant result in Azalea-unikernel.

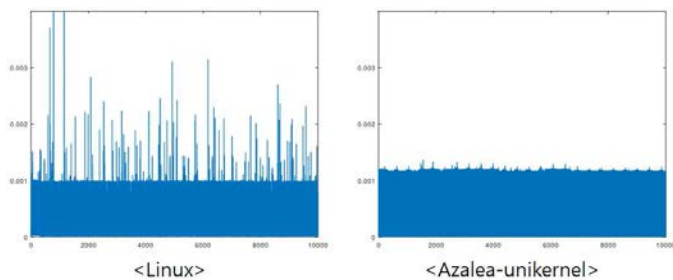


Figure 4. OS noise comparison result using FWQ benchmark

Figure 4 shows the result of running FWQ on Linux and Azalea-unikernel. X-axis represents the sample number, Y-axis represents the time taken for each sample in milliseconds. Both Linux and Azalea-unikernel show low noise, however, in the case of Linux it can be seen that spikes are periodically high. This is because Azalea-unikernel does not run unnecessary asynchronous kernel events or daemons.

As a result of FTQ, FWQ, and Hourglass benchmark experiments, we found that the OS noise is improved in Azalea-unikernel. In Azalea-unikernel, the results are constant, whereas, frequent spikes can be seen in Linux. The spike means that the application does not fully utilize the CPU computation power, which means that OS noise has occurred.

V. CONCLUSIONS

OS noise is a well-known problem in the HPC community, and many studies are conducted to solve it. Existing studies have also provided important insights on OS noise, and accordingly, related experiments should be conducted in the newly designed kernel.

In this paper, we verified the OS noise on the unikernel by comparing it with a general-purpose OS. We compared OS noise using Linux and Azalea-unikernel with well-known noise measurement benchmarks such as FTQ, FWQ, and Hourglass. Through experimental results using benchmarks, it was confirmed that the noise of Azalea-unikernel is smaller than that of Linux. This means that there is less kernel

interference on Azalea-unikernel, and it also means it is a kernel suitable for supporting scalability on manycore systems.

ACKNOWLEDGMENT

This work was supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2014-3-00035, Research on High Performance and Scalable Manycore Operating System)

REFERENCES

- [1] P. Beckman, K. Iskra, K. Yoshii, S. Coghlan, and A. Nataraj, "Benchmarking the effects of operating system interference on extreme-scale parallel machines," *Cluster Comput.*, vol. 11, no. 1, pp. 3-16, 2008.
- [2] Lameter, C. "Shoot first and stop the OS noise." In *Linux Symposium*, pp 159-168, 2009.
- [3] C. Engelmann, "Investigating operating system noise in extreme-scale high-performance computing systems using simulation," *IASTED Multiconferences - Proc. IASTED Int. Conf. Parallel Distrib. Comput. Networks, PDCN 2013*, pp. 670-677, 2013.
- [4] S. Wächtler and M. Sojka, "Operating System Noise: Linux vs . Microkernel," *Proc. RTLWS*, 2012.
- [5] K. B. Ferreira, P. Bridges, and R. Brightwell, "Characterizing application sensitivity to OS interference using kernel-level noise injection," *2008 SC - Int. Conf. High Perform. Comput. Networking, Storage Anal. SC 2008*, pp. 1-20, 2008.
- [6] M. Giampapa, T. Gooding, T. Inglett, and R. W. Wisniewski, "Experiences with a lightweight supercomputer kernel: Lessons learned from Blue Gene's CNK," *2010 ACM/IEEE Int. Conf. High Perform. Comput. Networking, Storage Anal. SC 2010*, 2010.
- [7] B. Geroft et al., "Performance and scalability of lightweight multi-kernel based operating systems," *Proc. - 2018 IEEE 32nd Int. Parallel Distrib. Process. Symp. IPDPS 2018*, pp. 116-125, 2018.
- [8] S. Lankes, S. Pickartz, and J. Breitbart, "HermitCore - A unikernel for extreme scale computing," *Proc. 6th Int. Work. Runtime Oper. Syst. Supercomput. ROSS 2016 - conjunction with HPDC 2016*, 2016.
- [9] S. H. Jeon, S. J. Cha, Ramneek, Y. J. Jeong, J. M. Kim, and S. Jung, "Azalea-Unikernel: Unikernel into Multi-kernel Operating System for Manycore Systems," *9th Int. Conf. Inf. Commun. Technol. Converg. ICT Converg. Powered by Smart Intell. ICTC 2018*, pp. 1096-1099, 2018.
- [10] Y. Jeong, J. Kim, S. Jeon, S. J. Cha, Ramneek, and S. Jung, "Design and Implementation of Azalea unikernel file IO offload," *ICTC 2019 - 10th Int. Conf. ICT Converg. ICT Converg. Lead. Auton. Futur.*, pp. 398-401, 2019.
- [11] A. Morari, R. Gioiosa, R. W. Wisniewski, F. J. Cazorla, and M. Valero, "A quantitative analysis of OS noise," *Proc. - 25th IEEE Int. Parallel Distrib. Process. Symp. IPDPS 2011*, pp. 852-863, 2011.
- [12] M. Sottile and R. Minnich, "Analysis of microbenchmarks for performance tuning of clusters," *Proc. - IEEE Int. Conf. Clust. Comput. ICCD*, no. 04, pp. 371-377, 2004.
- [13] Lawrence Livermore National Laboratory . The FTQ/FWQ Benchmark
- [14] Advanced Simulation and Computing, Lawrence Livermore National Laboratory. ASC Sequoia
- [15] E. A. Leon, I. Karlin, and A. T. Moody, "System Noise Revisited: Enabling Application Scalability and Reproducibility with SMT," *Proc. - 2016 IEEE 30th Int. Parallel Distrib. Process. Symp. IPDPS 2016*, pp. 596-607, 2016.
- [16] Collin B. McCurdy and Jeffrey S. Vetter. "Understanding the Behavior of the FWQ System Noise Microbenchmark." Technical Report FTGTR-2008-10, Oak Ridge National Laboratory, Future Technologies Group, 2008.
- [17] Regehr, J. "Inferring Scheduling Behavior with Hourglass." In *USENIX Annual Technical Conference, FREENIX Track*, pp. 143-156, 2002.



Seung-Jun Cha received his B.E., M.E. and Ph.D. degree in Computer Engineering from Chungnam National University, Daejeon, Korea in 2006, 2008, and 2013. Since 2013, he joins the Electronics and Telecommunications Research Institute (ETRI) where he is a senior researcher of Basic Technology Research Center for Next-Generation OS. His research interests include operating system principles include multi-kernel, microkernel, lightweight kernel, and unikernel for the manycore systems.



Seung Hyup Jeon received the MS degree from Korea University. He currently focuses on the operating system scalability in a manycore system. His research interests include system software for parallel computing, virtualization, and multi-tier memory system.



Yeon Jeong Jeong received the BS and MS degrees in Computer Science from Pusan National University, Pusan, Korea in 1994 and 1996, respectively, and the Ph.D. degree in Computer Science from Chungnam National University, Daejeon, Korea in 2005. Currently, he is a principal researcher in the Basic Technology Research Center for Next-Generation OS at ETRI, Daejeon, Korea. His research interests include unikernel, lightweight kernel, and multi-kernel for the manycore system.



Jin Mee Kim received her B.E. and M.E. degree in Computer Science from Pusan National University in 1988 and Chungnam National University in 1999 respectively. Since 1988, she joins the Electronics and Telecommunications Research Institute (ETRI) where she is a principal researcher of Basic Technology Research Center for Next-Generation OS. Her research interests include high-performance computing and operating system principles for the manycore systems.



Sungin Jung received his B.E., M.E. degree in Computer Engineering from Pusan National University in 1987, 1989, and Ph.D. degree in Computer Engineering from Chungnam National University, Daejeon, Korea in 2006. Since 1990, he has 28 years of experience in the area of the operating system and was a UNIX kernel developer during the 1990s. He had participated in developing the UNIX kernel for SMP, ccNUMA, and MPP systems in cooperation with Novell and SCO. In the early 2000s, he began the Linux kernel project for CGL (Carrier Grade Linux) and DCL (DataCenter Linux) workgroups of Linux Foundation. Since 2014, he is the director of Basic Research Center for manycore OS research. In addition, he is serving OSS activities such as OSS policy and international events. His research interests are operating system kernel, cloud computing, intermittent computing, and OSS.