International Journal of Computer Science and Information Technology
ISSN: 3005-9682 (Print), ISSN: 3005-7140 (Online) | Volume 7, Number 2, Year 2025
DOI: https://doi.org/10.62051/ijcsit.v7n2.07
Journal homepage: https://wepub.org/index.php/IJCSIT/index

Performance Optimization of Airport Control Area Pass System Based on Huawei Kunpeng

Haonan Zhang *, Tie Cao, Liming Shao, Hao Tu

The Second Research Institute of CAAC, Chengdu, Sichuan, 610000, China
*Corresponding Author: Haonan Zhang (E-mail: zhn513022@163.com)

ABSTRACT

At present, there are many problems in the supervision and industry services of personnel in airport control areas, mainly focused on data governance capabilities. Based on this pain point, develop a national airport control area pass personnel management system to achieve online application, approval, issuance, replacement, and reissuance of airport control area passes, achieve unified supervision and standardized processes, and comprehensively enhance safety supervision and management capabilities and data sharing capabilities. Compared to traditional X86 architecture, processors based on ARM architecture provide higher memory bandwidth. The pass system was successfully ported to the Huawei Kunpeng processor based on ARM architecture for testing and performance comparison with x86 architecture. The results showed that Huawei Kunpeng had significant performance optimization in both single core and multi-core computing efficiency compared to x86 architecture. The rapid development of domestic chip technology is fully empowering the digitalization of the civil aviation industry.

KEYWORDS

Control area pass; ARM architecture; Performance optimization; Huawei Kunpeng

1. INTRODUCTION

With the rapid development of China's civil aviation industry and the continuous expansion of the number and scale of airports, the management and security supervision of personnel holding airport control area passes are facing unprecedented challenges. The safety management of such personnel is a key link in aviation security work. Strengthening the supervision of this group is not only a core measure of the Civil Aviation Administration of China's Public Security Bureau to maintain aviation security but also an effective way to promote the implementation of intelligent supervision and improve the service level of the industry.

Based on the above requirements, a management system for personnel holding airport control area passes has been developed, which realizes functions such as online application, approval, issuance, renewal, and cancellation of control area passes, and supports electronic signatures and paperless archiving.

In the civil aviation software industry, x86 architecture processors are widely used. With the popularization of cloud computing, ARM architecture has begun to be applied in cloud computing platforms due to its advantages in energy efficiency. Domestic company Phytium Information Technology Co., Ltd. has launched the Phytium server-level processor based on ARM architecture [1]. Huawei Technologies Co., Ltd. has independently developed the Kunpeng series of server processors, which also adopt ARM architecture and feature high performance, high throughput, high

integration, and high energy efficiency. Xue Chunhui et al. have optimized DPDK on the Kunpeng server [2], and Chen Yefeng et al. have transplanted WRF to the Kunpeng processor and conducted evaluations [3].

2. CONTROL AREA PASS SYSTEM AND SERVER RESOURCES

2.1. Control Area Pass System

The control area pass system revolves around the management of the entire lifecycle of airport personnel, vehicles, and in-field driving licenses, achieving digitalization, intelligence, and self-service throughout the entire process of certificate application, review, payment, training, examination, certificate production, inspection, recycling, management, and supervision. It meets the needs of personnel applying for airport control area passes, application units, management units, and supervision units, as shown in Figure 1.

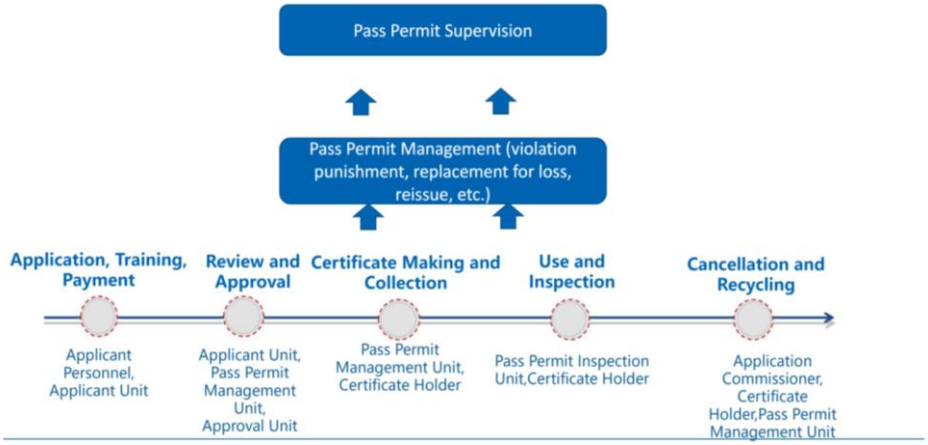

Figure 1. Pass Lifecycle Management

2.2. Server Resources

Kunpeng 920 is a server-grade processor launched by Huawei, based on the ARMv8.2 architecture and manufactured using a 7nm process technology. Its microarchitecture is Huawei's self-developed Taishan v110 microarchitecture [4]. The Kunpeng 920 processor features a highly integrated design, incorporating multiple computing core structures within a single chip. It offers various configuration options, including 32-core, 48-core, and 64-core configurations, and integrates an I/O control unit. Equipped with an eight-channel DDR4 SDRAM memory controller, this processor significantly enhances data throughput capacity and supports Non-Uniform Memory Access (NUMA) architecture.

Deploy the control area pass certificate onto a server equipped with a Kunpeng920 processor. The comparison of its configuration and resource with a server equipped with an X86 architecture is shown in the table below.

Table 1. Hardware Comparison

Architecture	X86	ARM
Product Model	Intel Xeon	TaiShan 200 Server 2280
Processor	E5-2698	Kunpeng920
Memory	16G	16G
Hard Disk	500G	200G
Network Card	1*4*GE	1*4*GE

Kunpeng platform provides two compilers: GCC compiler and Bi Sheng compiler [5] When facing high-performance computing (HPC) applications, Kunpeng platform prioritizes using the Bi Sheng compiler as its core compilation tool This compiler is built on the open-source LLVM framework and focuses on providing compilation support for C, C++, and Fortran languages In the specific implementation, clang in the LLVM infrastructure is used as the compilation driver for C/C++, while flang is responsible for the compilation and driver processing of Fortran language.

In addition to inheriting the general functions and optimization capabilities of LLVM, the Bi Sheng compiler has also implemented deep enhancements for key technical links in the mid-range and backend processes Meanwhile, the compiler integrates an Auto tuner, which enables automated optimization and configuration of compilation options This feature enables it to significantly improve program performance in HPC applications.

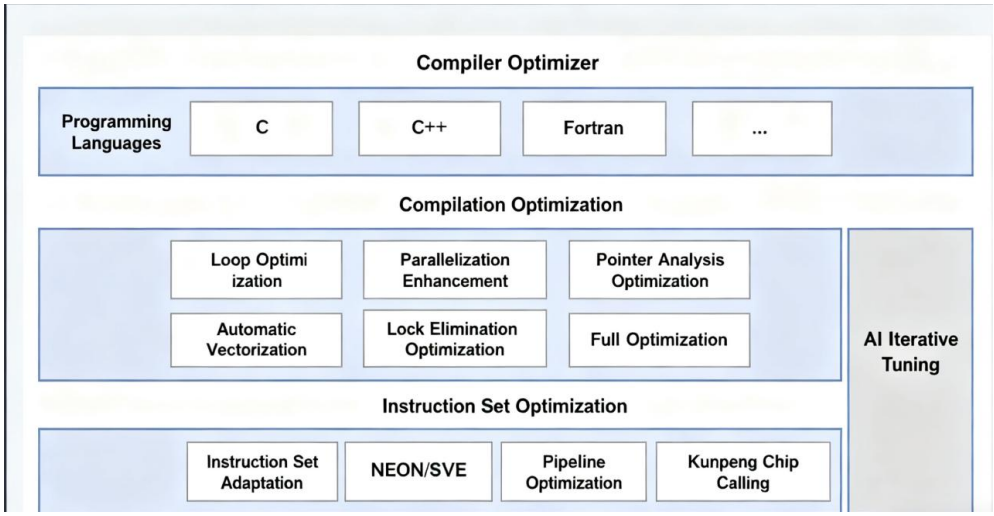

Figure 2. Bi Sheng Compiler Architecture

3. PERFORMANCE TESTING

3.1. Single Core Performance

The primary step in performance tuning lies in constructing an efficient basic operating environment. Initially, this project adopted OpenJDK 11, and its benchmark tests revealed a significant bottleneck in single-core processing performance. To fully unleash the potential of the underlying hardware, the runtime environment was upgraded to Oracle JDK 17.

In the JDK17 environment, the testing time for data processing on a single core of the x86 architecture is 4866ms, while the time on a single core of the Kunpeng architecture is 2522ms, representing a performance improvement of 42.86% compared to x86.

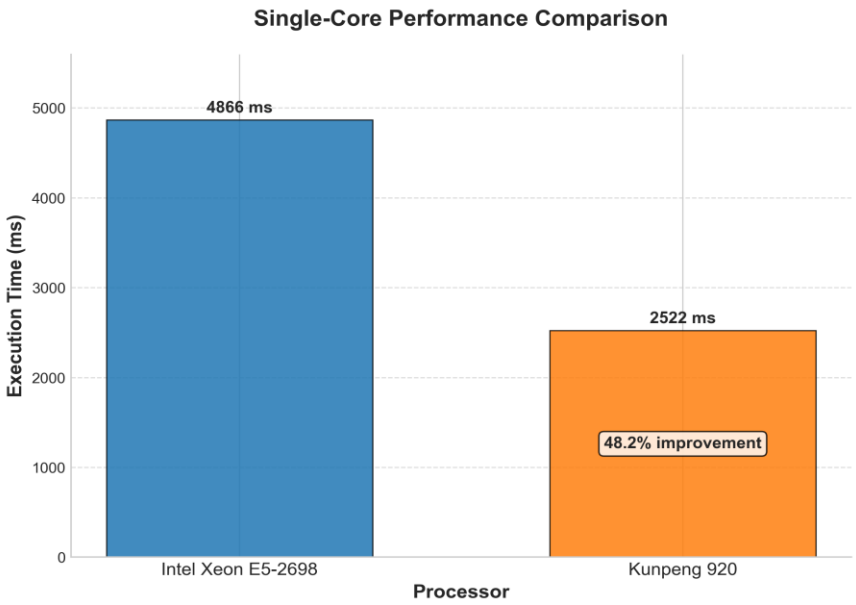

Figure 3. Single Core Performance Comparison

3.2. ForkJoin Thread Pool Performance

To completely solve the performance constraints of computationally intensive tasks, this study restructured the core algorithm, adopted a Divide and Conquer strategy, and introduced the Java native ForkCoin framework for implementation.

Based on the principles of Divide and Conquer algorithm design, this study restructured the core algorithm and introduced the Java native ForkCoin framework for implementation The core advantage of this framework lies in its efficient Work Stealing scheduling algorithm, which can effectively balance the load among multi-core processors, greatly reduce thread idle and synchronous waiting overhead, and is particularly suitable for handling a large number of fine-grained asynchronous tasks [6].

After optimization, the processing time of Kunpeng and X86 platforms decreased from 2522ms and 4866ms to 27ms and 30ms, respectively. The time consumption of the two platforms tended to be consistent (27ms vs 30ms). This confirms the recent research conclusion that the ForkCoin framework based on work stealing mechanism can effectively improve the utilization of computing resources in multi-core environments, especially exhibiting excellent cross platform performance in data intensive application scenarios.

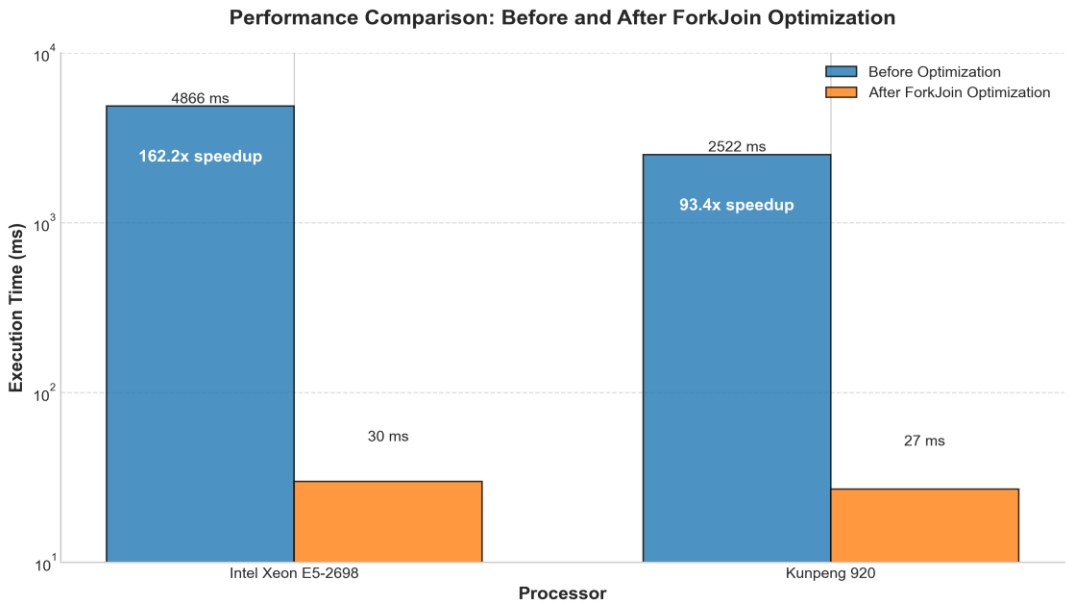

Figure 4. Comparison of ForkJoin Optimization Performance

3.3. Multi Core Performance

By dynamically obtaining the number of available processors through the availability Processor function, and explicitly setting the thread pool size cores to 8 based on this, the number of executing threads is ensured to be consistent with the number of physical computing cores. This aims to eliminate unnecessary thread context switching overhead and avoid processor idle caused by too few threads or resource contention caused by too many threads.

The precise tuning of thread pools has brought significant parallel acceleration benefits: (1) Near linear acceleration ratio: In an 8-core environment, X86 and Kunpeng platforms achieved acceleration ratios of 7.37 and 7.67 respectively (calculated as single core time/multi-core time), highly approaching ideal linear acceleration (theoretical value of 8) This proves that the task is well decomposed and the inter thread synchronization overhead is controlled at an extremely low level (2) Platform performance consistency: The multi-core performance of Kunpeng platform (329ms) is nearly twice that of X86 platform (660ms), which is consistent with the difference in single core performance between the two (2522ms vs 4866ms), indicating that concurrent optimization strategies can effectively unleash their hardware potential on different architectures.

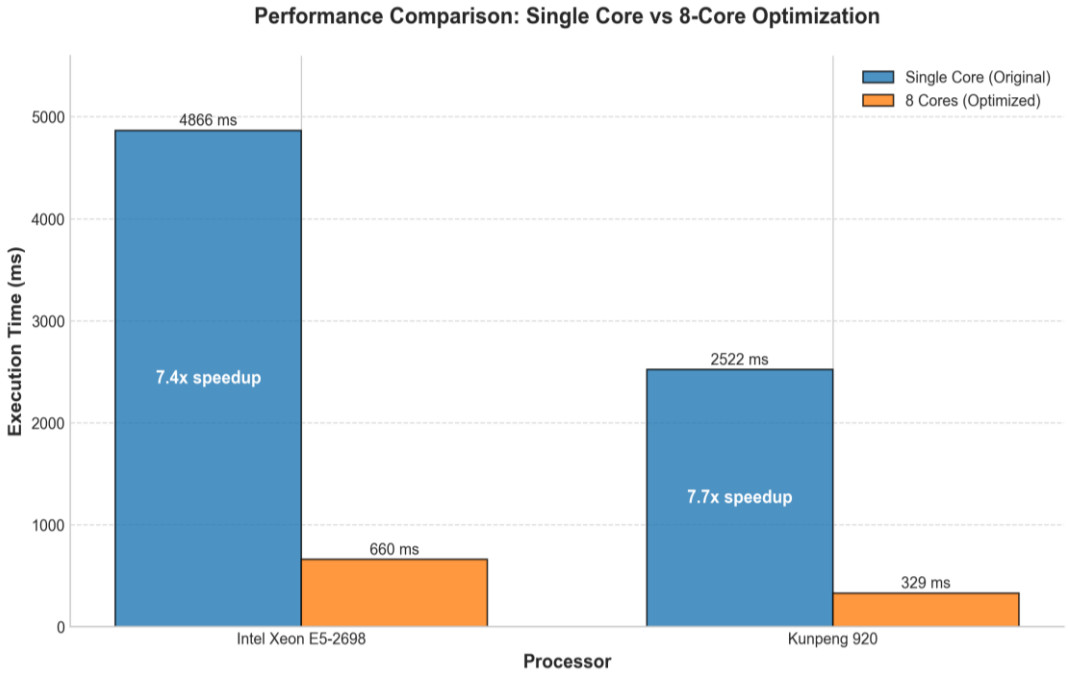

Figure 5. Comparison of Multi Core Computing Performance

4. SUMMARY

This study systematically optimized the performance of the airport pass management system through three stages The experimental results show that firstly, by migrating the runtime environment from OpenJDK 11 to Bi Sheng JDK 17, which is deeply optimized for ARM architecture, the single core performance of Kunpeng 920 processor is significantly improved, and the time consumption is reduced by about 58%, proving the importance of software hardware collaborative optimization Secondly, by introducing the ForkCoin parallel computing framework to reconstruct the core algorithm, the most significant optimization effect was achieved, reducing the processing time of both platforms by two orders of magnitude and improving performance by about 160 times. This highlights the key role of advanced parallel algorithms in unleashing the performance potential of multi-core processors Finally, by accurately aligning the concurrency of the thread pool with the number of physical CPU cores, a nearly linear acceleration ratio (7.37 times and 7.67 times) was achieved, ensuring optimal utilization of system resources This solution successfully improves the system's response from seconds to milliseconds, providing effective support for real-time processing in high concurrency scenarios.

Overall, porting the pass management system to the Kunpeng 920 processor platform can achieve reliable performance improvements, and in the future, other civil aviation software systems can be ported and optimized based on the Kunpeng platform.

REFERENCES

[1] Feiteng Information Technology Co., Ltd Feiteng Tengyun S series high-performance server CPU [EB/OL]. [223-3-20] https://www.phytium.com.cn/homepage/production/list/0

[2] Xue Chunhui, Yu Liangwen, Li Jia, etc Optimization of DPDK on ARM architecture domestic servers [J]. Network Security Technology and Applications, 2023 (10): 1-4.

[3] Chen Yefeng, Yan Chen, Chen Feng, etc WRF porting and evaluation based on Kunpeng processor [J]. Frontiers of Data and Computing Development, 2024, 6 (3): 150-161.

[4] WikiChip, TaiShan v110-Microarchitectures-HiSili con-HiSilicon for Huawei's own TaiShan servers [EB/OL]. https://en.wikichip.org/wiki/hisilicon/microarchitectures/taishan_v110

[5] Zhu Wenjin, Bao Xiangwen, Wang Yuliang, etc A Bi Sheng compiler function inline optimization method based on Kunpeng platform, storage medium, device, and computer program product: 202410651492 [P] [2025-08-29].

[6] Li Hua, Wang Ming, Zhang Qiang Research on Parallel Computing Optimization Based on Fork/Join Framework [J]. Computer Engineering and Applications, 2021, 57 (15): 76-82.