Porting LHAASO WFCTA simulation job to ARM computing cluster

Yaosong Cheng[1,*], *Yujiang* Bi[1,2,**], *Jingyan* Shi[1], *Chaoqi* Guo[1], *Xiaofei* Yan[1]

[1]Institute of High Energy Physics, CAS, 100049 Beijing, China
[2]Tianfu Cosmic Ray Research Center, Institute of High Energy Physics, Chinese Academy of Sciences, 610041 Chengdu, China

Abstract. With the advancement of many large-scale high-energy physics experiments, the amount of data to be processed and analyzed has significantly increased. For example, since the start of the Large High Altitude Air Shower Observatory (LHAASO) experiment in 2020, their simulation jobs have been running on an Intel X86 cluster, producing only a fraction of the planned data for the first phase due to limited CPU resources. Therefore, it is necessary to explore and expand other computing service devices. We built an application ecosystem based on the ARM architecture to support offline data processing for high-energy physics. The main work includes porting the offline software based on LHAASO experiments to run on ARM machines, formulating data transfer and job scheduling strategies in the ARM cluster, and evaluating performance and power consumption in both Intel X86 and ARM clusters. The results show that the LHAASO simulation jobs can run correctly on the ARM computing cluster. The single-core performance of Intel X86 CPUs is better than ARM CPUs, but for the entire server with a multicore architecture, ARM servers perform better.

1 Introduction

The Large High Altitude Air Shower Observatory (LHAASO) [1] is a large cosmic ray observation array located at an altitude of 4,410 meters in Daocheng County, Sichuan Province, China. It consists of a hybrid array of approximately 1 square kilometer, including a 1.3 square kilometer Electron Antiparticle Scattering Array (KM2A), a 78,000 square meter Water Cherenkov Detector Array (WCDA), and 12 Wide Field of View Cherenkov/Fluorescence Telescopes (WFCTA), as shown in Figure 1. It is the world's highest-altitude, largest-scale, and most sensitive cosmic ray detection device, and is one of China's major national science and technology infrastructures.

* e-mail: chengys@ihep.ac.cn
** e-mail: biyujiang@ihep.ac.cn

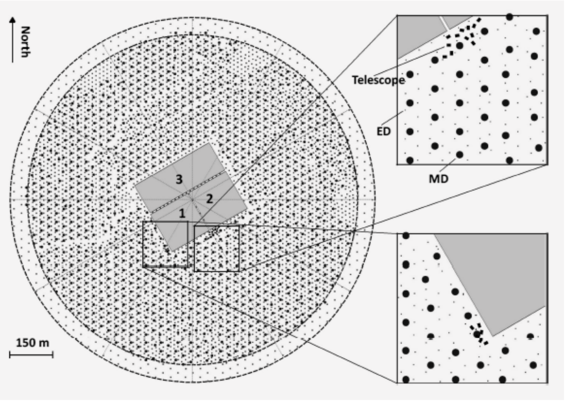

Fig. 1. Distribution map of the three types of detectors in LHAASO

The construction of the LHAASO experimental facility began in 2018, and data collection and operation of data analysis experiments and simulation experiments started in 2020. So far, approximately 150 million jobs have been completed, consuming around 130 million CPU hours. Currently, the experiment runs approximately 360,000 jobs per day, consuming 350,000 CPU core hours, and the daily data collection and generation reach 20TB. In order to meet the experimental requirements, we have established a large-scale computing cluster with a total of approximately 15,000 CPU cores across different sites, along with 50PB of disk storage and 40PB of tape storage. However, there is still a resource gap, and many jobs require long queues to proceed. For example, the simulation experiment of WFCTA, which started in 2020, has only completed about 25% of the target for producing and analyzing simulated data. Therefore, under limited conditions, we urgently explore more cost-effective computing solutions.

To address this, we explored the porting of high-energy physics data simulation software Corsika [2] and G4KM2A [3] from Intel x86_64 to ARM aarch64 architecture, and conducted performance testing. We selected Corsika-V77420 and G4KM2A-10.1 versions for porting, as they were stable versions commonly used by physicists when we started this work. Section 2 of the article briefly introduces the ARM architecture and its application in high-energy physics experiments. Section 3 describes our porting work and the main challenges encountered. The performance test results and discussions of the software will be presented in Section 4. Finally, we summarize our findings in section 5.

2 ARM architecture and its role in HEP experiments

ARM is a widely used Reduced Instruction Set Computing (RISC) architecture that was initially developed by the British company ARM Holdings. It has now become one of the most widely used embedded processor architectures globally. ARM-based CPUs are widely used in smartphones, laptops, tablets, and embedded systems. Compared to Intel's x86 architecture, ARM has the advantages of being smaller in size, having lower power consumption, and lower cost. These characteristics give ARM a certain advantage in high-performance computing (HPC) applications that require significant power costs. With the release of the new ARMv8 architecture, more and more manufacturers are starting to develop ARM-based server-grade chips and apply them in distributed storage, big data analytics, cloud computing, and other scenarios. Examples include the Cavium ThunderX [4] from the United States, the Fujitsu A64FX [5] from Japan, and the Huawei Kunpeng 920 [6] from China. At the same time, researchers have also explored the performance differences between ARMv8 and x86 series from different perspectives.

Laurenzano et al. tested and evaluated the performance of the first commercially available 64-bit ARMv8 architecture processor, X-Gene [7]. The control group included the commonly used Intel E5-2670v1 (Sandy Bridge) processor in HPC systems and a typical low-performance, low-power Atom C2758 processor. The research results showed that the performance of X-Gene was slightly higher than that of the Intel Atom chip, and the power consumption was slightly lower than that of the Intel Sandy Bridge chip, mainly due to the design differences in the memory subsystem. In recent years, with the continuous development of the ARMv8 architecture, it has also been better optimized and applied in the HPC field.

On the other hand, the field of high-energy physics, as a classic application scenario of high-performance computing, also pays great attention to the software ecosystem based on the ARM architecture. Many commonly used software in the field have been ported from x86 to aarch64, such as LHCb, LCG, and ATLAS software packages for international high-energy physics experiments. These efforts are of great significance to the software development ecosystem in the field of high-energy physics. Some representative cases, such as Marek et al., ported the LCG software suite to aarch64 and then used the "multicore/mtbb201_parallelhistfilll.c" program in ROOT software to test performance [8]. The research results showed that although the performance of single-core ARM machines was not as good as x86_64 machines, in multi-core benchmark tests, as the number of cores increased, the CPU time consumed by x86 and ARM tended to be the same. The advantage of ARM processors lies mainly in their cost-effectiveness rather than high performance. Additionally, Laura Promberger et al. ported the LHCb software suite from x86 to aarch64 and ppc64le architectures [9]. They also analyzed and demonstrated the importance of cross-platform support for vectorization. These software porting efforts and achievements have inspired us to port the Corsika and G4KM2A software required for the LHAASO survey task to the aarch64 architecture.

3 ARM-based computing cluster ecosystem for HEP

The software porting work is implemented based on the Big Science Intelligent Computing Center located in Dongguan, Guangdong Province, China. It provides a total of 30,000 CPU cores, including 9,600 cores based on the 64-bit ARMv8 architecture. In addition to high-performance and high-throughput computing clusters, we have also built a complete set of high-energy physics computing environments, including login clusters, distributed storage systems, and management clusters, all interconnected by a 100Gb RoCE network.

Before testing software performance, we first establish a high-energy physics data analysis environment based on ARM servers and port some general or domain-specific software, such as the computation platform software HTCondor [10] and Slurm [11], distributed storage software EOS, and high-energy physics data analysis software ROOT and GEANT [12]. In addition, to make the job submission process more convenient, we have developed a job scheduling strategy across heterogeneous computing clusters, as shown figure 2.

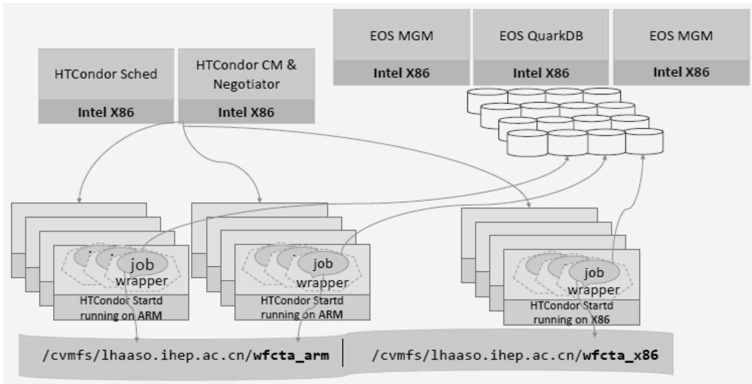

Fig. 2. Job scheduling strategy of the cluster

When a user submits their simulation job on the login node, they do not need to worry about which type of working node the job will run on. Instead, it is determined by the scheduler. The scheduling strategy is based on priority, distribution of available resources, and other factors. After selecting a working node, the job wrapper packages the user's job script and detects the hardware architecture. It then selects the appropriate software version and loads the corresponding environment variables from the public software distribution platform CVMFS based on the hardware architecture type. For example, in the figure, the HTCondor scheduler sends the user's job to either an ARM node or an X86 node. After the job is packaged, the wrapper queries the processor architecture of the current node to choose different software versions to load, including ROOT, Corsika, GEANT4, and other software runtime environment variables. Once everything is ready, the data is called from the EOS storage system to complete the data analysis. In addition, the scheduler also supports users specifying the hardware architecture for job execution, but there may be a waiting time during busy periods of the cluster.

4 Performance comparison between ARM and x86

The ARM server model used in the Intelligent Computing Center is the Huawei TaiShan 200K series high-performance computing server, equipped with the Kunpeng 920 CPU. The specific architecture is aarch64 ARM v8.2. It has three different configurations: 32, 48, and 64 cores. The server has two 48-core CPUs, totalling 96 cores, with a clock frequency of 2.6GHz. The more detailed configuration information of the server and the versions of the ported software are shown in the table below:

Table 1. Font styles for a reference

Item	Parameter
CPU	HiSilicon Kunpeng 920-6426 CPU @ 2600MHz 64cores * 2EA
network card	25 Gbps
Disk drives	SSD drives with SAS interface (SATA 3.2, 6.0 Gb/s)
Operating system	CentOS Linux release 7.6.1810
Kernel	4.14.0-115.10.1.el7a.aarch64
HTCondor version	9.1.0
CVMFS version	2.5.2
ROOT version	6.20
CORSIKA version	V77420

Based on these configurations, we conducted several tests, including the HEP-SPEC06 benchmark test [13], WFCTA simulation job test, and WCDA example reconstruction test. While running the test programs, we also monitored the power consumption of the server because low power consumption is one of the advantages of ARM processors, and it is necessary to evaluate it.

4.1 HEP-SPEC06 Benchmark Test

HEP-SPEC06 (HS06) is a standard tool for evaluating CPU performance in the field of high-energy physics. The control group includes three X86 processors: the recently released AMD EPYC 7773x, with 64 cores and a clock frequency of 2.20GHz per CPU, totaling 128 cores in the entire server; the commonly used Intel Xeon 8352Y, with 32 cores and a clock frequency of 2.20GHz per CPU, totaling 64 cores in the entire server; and the Intel Xeon 6258R, which has a similar clock frequency to Kunpeng ARM, with 28 cores and a clock frequency of 2.70GHz per CPU, totaling 56 cores in the entire server.

We separately tested the single-core score and the overall server score when running a single example. The overall server test refers to the score when running examples equal to the number of CPU cores in the server. The results are shown in the figure below:

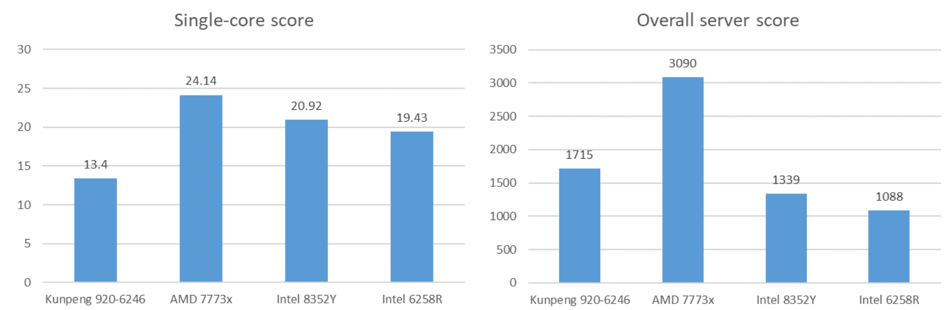

Fig. 3. Single-core score and the overall server scores

The results show that the latest AMD X86 chips achieved the highest scores in both single-core and overall performance. The ARM processor scored lower in single-core performance compared to the three X86 processors, but it outperformed the 8352Y and 6258R when evaluating the overall server performance. While conducting the benchmark test on the entire server, we also monitored the power consumption using the "ipmitool" tool for statistics. The results are shown in the figure below:

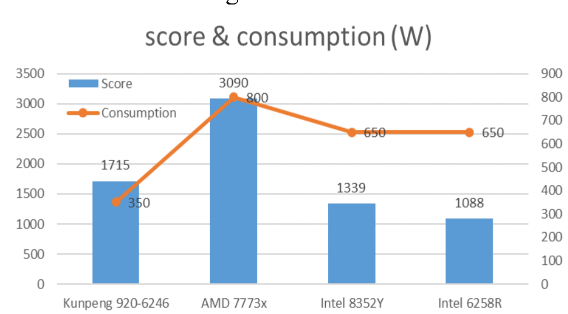

Fig. 4. Overall server score and power consumption

In figure 4, it can be seen that the ARM server has the lowest power consumption. To better evaluate the relationship between power consumption and performance, we used the score-to-power ratio, which is calculated as score divided by power consumption. In figure 5, it can be seen that ARM has the highest score-to-power ratio.

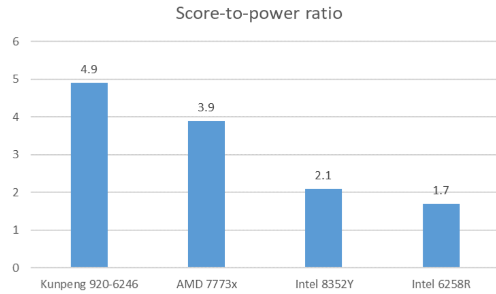

Fig. 5. Score-to-power ratio

4.2 WFCTA Simulation Job Test

Next, we conducted tests using real WFCTA simulation jobs. The Corsika software, using the GHEISHA model, was used for the tests. Similar to the HS06 test, we used the server with Kunpeng 920 as the test platform. The control group included the Intel Xeon 6258R and a server with a similar price to ARM, equipped with two Intel 5218 CPUs with a total of 32 cores.

Running a Corsika simulation job occupies one CPU core. We tested the time and power consumption for both single-core and the entire server CPU cores to complete the job on all servers. The results of the single-core test are shown in the table below:

Table 2. Single-core test

CPU type	Number of jobs	Running time (m)	Idle consumption (W)	Operating consumption (W)	Csonsumption svariation (W)
ARM-920	1	109	300	306	**+6**
X86-6258R	1	96	110	160	**+50**
X86-5218	1	90	180	240	**+60**

As we can see, the same Corsika simulation job takes an average of 109 minutes to complete on the ARM server, while it takes 96 and 90 minutes on the X86 servers, respectively. In terms of single-core performance, the Kunpeng 920 is slightly weaker than the two X86 CPUs. In terms of power consumption, since there are significant differences in hardware configurations among the servers, we are more concerned about the difference in power consumption between when running the job and when idle. From the rightmost column in the table, we can see that the power consumption of the ARM server only increases by 6W when running a Corsika simulation job, while the power consumption of the X86 servers increases by more than 50W. Therefore, it can be concluded that the energy consumption of the ARM-based Kunpeng 920 CPU for running jobs is much lower than that of the commonly used X86 CPUs.

To evaluate the time and power consumption of running Corsika simulation jobs on servers under full load, we conducted another set of experiments. We ran simulation jobs on each test server with the same number of CPU cores and measured the power consumption when the servers were under full load. The results are shown in the Figure 6 :

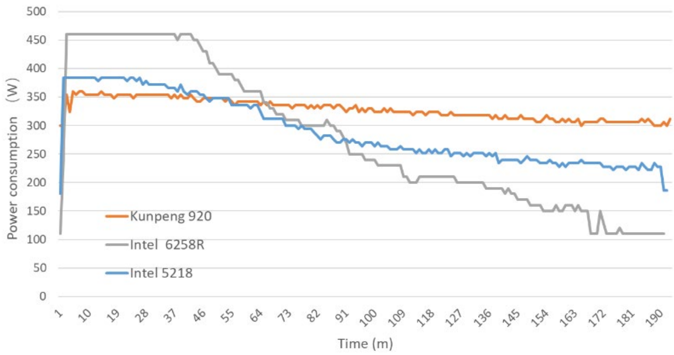

Fig. 6. Power consumption variation of the server during Corsika job execution

In this test, although each CPU core is responsible for processing one job, there are still some resources that need to be competed for, such as memory, storage, file systems, etc. This leads to some jobs being able to finish quickly after being launched, while others take a long time to complete due to waiting for resource release. Therefore, we consider the time from when the jobs start running until all jobs are completed as the total running time.

It can be seen that the Kunpeng 920 took the longest time to complete all computational tasks, with an average time of 103.75 minutes per job, while the fastest Intel 6258R only took an average of 77.17 minutes per job. It can be concluded that the two X86 CPUs have better single-core computing capabilities than the ARM-based Kunpeng 920.

In terms of power consumption, we also used the "ipmitool" [14] to measure the power consumption during the running phase of all jobs. From the figure, it can be seen that the Kunpeng 920 has higher power consumption when idle, but the power consumption variation during running the simulation jobs is much smaller than that of the two X86 processors.

To evaluate the energy consumption generated by running Corsika simulation jobs on the test servers, we used the difference between the final power consumption and the product of idle power consumption and running time as the evaluation metric. From the table 3, it can be seen that the ARM server based on the Kunpeng 920 architecture consumed only 125.51 watt-hours of electricity when running 96 simulation jobs, while the X86 servers were 3 to 5 times higher (614.57 and 367.36 watt-hours), with fewer jobs running concurrently than the ARM server. These experimental results indicate that the ARM server based on the Kunpeng 920 architecture has sizeable power consumption advantages when running Corsika simulation jobs.

Table 3. Overall server test

CPU type	Number of jobs	Running time (m)	Average running time per job	Electricity consumption (WH)	Jobs electricity consumption (WH)
ARM-920	96	4h6m	103.75m	1355.51	125.51
X86-6258R	56	2h54m	77.17m	933.57	614.57
X86-5218	32	3h20m	83.09m	967.36	367.36

5 Conclusion

Our work expands the running scenarios of LHAASO simulation jobs, allowing them to run on both x86 servers and ARM-based devices, providing us with more hardware options when choosing. Throughout the research process, we first established a high-energy physics data analysis environment based on ARM servers and ported some general or domain-specific software, setting up job scheduling strategies for heterogeneous computing clusters. After setting up the running environment, we first tested the scores and power consumption of different architecture servers in the HEP-SPEC06 benchmark test. The results showed that the single-core score of the ARM-based Kunpeng 920 CPU was lower than that of the x86-based AMD-EPYC 7773x, Intel Xeon 8352Y, and 6258R. However, Kunpeng 920 had outstanding advantages in terms of score-to-power ratio under full load. Then, we tested the real Corsika simulation jobs of the LHAASO-WFCTA experiment, and the results showed that Corsika simulation jobs on ARM servers took more time to run. However, in both single-core and full load tests, the energy consumption of ARM servers for computational tasks was lower than that of the two x86 servers.

6 Acknowledgments

This work is supported by the National Natural Science Foundation of China (No.12075268), the Science and Technology Innovation Project of Institute of High Energy Physics, Chinese Academy of Sciences (No. E15451U2), and the Youth Innovation Promotion Association of CAS (No. 203013).

References

1. Y.D. Cheng, J.Y. Shi, G. Chen, Overview of High Energy Physics Computing Environment. Sci. Info. Tech. & Appl. **5(3)**, 3-10 (2014)
2. D. Heck, J. Knapp, J.N. Capdevielle, et al. CORSIKA: a Monte Carlo code to simulate extensive air showers. (1998)
3. S. CHEN S. Detector Simulation of LHAASO-KM2A with Geant4. PoS, **219,** (2020)
4. S.M. Smith, J. Price, T. Deakin, A. Poenaru, A performance analysis of the first generation of HPC - optimized Arm processors. CCPE, **31** (2019)
5. T. Yoshida, Fujitsu high performance CPU for the Post-K Computer. HCS (2018)
6. Z. Dai, J. Liu, Architecture and Programming of Kunpeng Processor, Tsinghua University Press, (2020)
7. M.A. Laurenzano, A. Tiwari, A. Cauble-Chantrenne, A. Jundt, L. Carrington, Characterization and bottleneck analysis of a 64-bit ARMv8 platform. ISPASS, **36**, (2016)
8. M. Marek, G. Ganis, P. Mato, I. Razumov, Cern students note, Porting the LCG software stack to the ARM architecture. (2019), https://cds.cern.ch/record/2690233
9. L. Promberger, M. Clemencic, B. Couturier, A.B. Iartza, N. Neufeld, Porting the LHCb Stack from x86 (Intel) to aarch64 (ARM) and ppc64le (PowerPC). EPJ, **214**, 05016 (2019)
10. M.J. Litzkow, M. Livny, M. Mutka. Condor-A Hunter of Idle Workstations. 1988 Proceedings. The 8th International Conference on Distributed. IEEE, 104-111 (1988)
11. M. Jette, C. Dunlap, J. Garlick. SLURM: Simple Linux Utility for Resource Management. (2002)
12. S. Agostinelli, J. Allison, K. Amako, et al. Geant4—a simulation toolkit. Nuclear Instruments and Methods in Physics Research Section A Accelerators Spectrometers Detectors and Associated Equipment, **3(506)**, (2003)
13. HEPSpec06 specifications, http://w3.hepix.org/benchmarks/
14. D.S. Wung. Intelligent Platform Management Interface (IPMI). Federation Proceedings, (2011)