
TECHNICAL WHITE PAPER

Debian 13 & Proxmox VE on Huawei TaiShan

Porting a production-ready operating system stack to the Kunpeng-based ARM64 platform, including the adaptation of hardware drivers to the Linux 7.x kernel.

1 Executive summary

Dawico Labs has brought Debian 13 “trixie” to Huawei's TaiShan server platform (Kunpeng, ARM64), both as a native bare-metal system and as a virtualization host under Proxmox VE. The focus was on adapting the hardware drivers to the Linux 7.x kernel.

The vendor's own `hiraidd` driver for the Huawei SP686C RAID controller is only intended for older kernel versions. On modern kernels from version 7.x onwards it would neither compile nor run, due to far-reaching ABI changes in the Linux kernel. Without adaptation the system did not start at all: the controller triggered a kernel panic during boot.

Through targeted refactoring of the driver sources, packaging within the DKMS system and integration into adapted installation media, the controller is now available for current enterprise systems on ARM64.

Result: The driver compiles and loads cleanly against the real target kernel; the RAID virtual disk is visible to the operating system and usable as the root drive. Debian 13 runs natively and under Proxmox VE on the same adapted kernel base.

2 The platform: Huawei TaiShan

TaiShan is Huawei's ARM-based server line, powered by the Kunpeng processor family. Many cores, high memory bandwidth and remarkable energy efficiency make the platform a foundation for cloud-native workloads, big data and distributed systems.

Characteristic	Details
Architecture	ARM64 (aarch64) with Huawei Kunpeng processors
Model range	1U, 2U and 4U rack servers plus high-density systems
Processors	1 to 4 sockets, up to 64 cores per CPU, SMT with two threads per core on current generations
Memory	DDR5 on current generations, eight and more memory channels per CPU
Storage	NVMe, SAS and SATA configurations with hot-swap
I/O & networking	PCIe 4.0 and newer; 10GE/25GE up to 100GE
Reliability	Redundant power supplies and fans (N+1), hot-swap components

Specifications vary by model and generation (manufacturer data). Representative models include TaiShan 1280 (high density, 1U), TaiShan 2280 (balanced), TaiShan 2480 (high-performance, 4 sockets) and TaiShan 5280 (storage).

3 Objective: native and virtualized

The operating system stack had to work in both operating modes that matter in practice, on an identical, adapted kernel base:

- **Native Debian 13 (bare metal).** A lean, minimal Debian straight on the TaiShan hardware, for server workloads, containers and cloud-native stacks, with no virtualization overhead.
- **Proxmox VE.** The TaiShan server as a virtualization host for VMs and containers on ARM64, managed through Proxmox VE, which itself builds on a Debian base.

Both paths share the same adapted kernel foundation. Every piece of driver work needed for native operation therefore benefits the virtualized variant directly.

4 The challenge: from kernel 6 to kernel 7

The hardware drivers in use were originally developed against a Linux 6 kernel. The first compilation attempt against the real headers of the target kernel revealed that the SCSI interface in particular had been reworked more extensively than expected. The following adaptations were verified empirically against the actual header tree, not estimated.

4.1 Overview of the kernel ABI changes

Area	Kernel 6.x	Kernel 7.x (verified)
Unaligned header	<code><asm/unaligned.h></code>	<code><linux/unaligned.h></code>
SCSI device callbacks	<code>slave_alloc/_destroy/_configure</code>	renamed to <code>sdev_init/_destroy/_configure</code>
Configuration callback	one parameter	additional <code>queue_limits</code> parameter
Block layer limits	dedicated setter calls	consolidated into the <code>queue_limits</code> API
BSG queue API	5 parameters	extended layout with 6 parameters

Version checks in the source code keep the driver backwards compatible with older kernels while allowing it to build natively against kernel 7.x.

4.2 The RAID controller: the biggest hurdle

At the heart of the problem was the storage/RAID controller. When its SCSI device callbacks were missing from the driver template (because they carry different names on kernel 7.x), the device-specific memory was not allocated during the SCSI scan. The first I/O command then dereferenced a null pointer, resulting in a kernel panic from within the initramfs, before the actual system start:

```
/* SCSI scan without active device callbacks: hostdata stays NULL */
hostdata = sdev->hostdata;
io_cmd.rw.hdid = cpu_to_le32(hostdata->hdid); // NULL deref → KERNEL PANIC
```

The solution: the affected callbacks remain fully active, merely under their new names for kernel 7.x, and the outdated block-layer calls inside the device configuration are migrated to the new `queue_limits` API. The device-specific memory is allocated correctly again and the panic no longer occurs.

4.3 Correct CPU detection

A further area of work was the reliable detection and initialization of the Kunpeng processors on the ARM64 platform. Only once the CPU topology is detected cleanly do the scheduler, interrupt distribution and the low-level drivers work together stably, the basis for productive operation. This includes SMT (simultaneous multithreading, the hyper-threading principle): current Kunpeng processors run two threads per core, and all logical processors are fully available to the operating system.

5 Operations and maintenance

A driver compiled once is not enough in production. Two building blocks make the solution durable and free of manual work.

5.1 Update safety through DKMS

The driver was packaged as a DKMS module. System and kernel updates rebuild the module automatically in the background for whichever kernel is active. This removes the manual maintenance effort that otherwise turns such special-purpose solutions into a risk: after a kernel update the system still boots reliably from its RAID volume.

5.2 Boot and installer integration

So that the operating system can be installed directly onto the RAID volume, the compiled module including its module index was integrated into the live environments and ramdisks of the installation media. The RAID array is available as a drive as soon as the installer boots and can be used as the root drive immediately, with no intermediate steps or driver injection.

6 Verification on real hardware

The adaptations were not simulated but compiled and loaded against the real kernel on the target server. The kernel module reports the correct kernel signature for the aarch64 target environment:

```
$ modinfo hiraidd.ko | grep vermagic
vermagic: 7.x SMP preempt mod_unload modversions aarch64
```

After loading the module, the kernel detects the controller and the RAID1 virtual disk is attached as a block device:

```
$ insmod hiraidd.ko
scsi host8: hiraidd
scsi 8:2:0:0: Direct-Access      HUAWEI  RAID1          C00
sd 8:2:0:0: [sdc] 1874335744 512-byte logical blocks: (960 GB/894 GiB)
sd 8:2:0:0: [sdc] Attached SCSI disk
```

Evidence: The controller's RAID1 virtual disk is visible and addressable. The system boots stably from the RAID volume; the kernel panic seen previously no longer occurs.

7 System compatibility

Because the adaptations target the generic kernel ABIs, the solution can be deployed across ARM64 architectures:

Platform	Status	Implementation
Proxmox VE 9.2 (ARM64)	Successfully verified	DKMS integration and adapted ISO installer
Debian 13 (ARM64)	Successfully verified	DKMS package and adapted netinst media; installation and boot from the RAID volume confirmed on real hardware
Ubuntu 26.04 LTS	Successfully verified	Adapted installer with the hiraidd driver built in

The installation images for Debian 13, Proxmox VE and Ubuntu 26.04 LTS are available for download at huawei.dawico.dev. The roadmap on the same site shows the state of the ports.

8 About Dawico Labs

Dawico Labs is the research and development unit of Dawico Systems. Its focus areas are highly scalable HA systems, web and cloud engineering, global app projects and AI systems with GPU power. What proves itself in the lab goes into production as a blueprint.

Contact: huawei.dawico.dev · dawico.dev

Huawei, TaiShan and Kunpeng are trademarks of Huawei Technologies Co., Ltd. Debian is a trademark of Software in the Public Interest, Inc. Ubuntu is a trademark of Canonical Ltd. Proxmox and Proxmox VE are trademarks of Proxmox Server Solutions GmbH. Linux is a registered trademark of Linus Torvalds. All other trademarks are the property of their respective owners. This document describes independent work by Dawico Labs GmbH and does not constitute an official statement by any of the companies named. Technical details without warranty; configuration and availability vary by model, generation and region. © 2026 Dawico Labs GmbH.