DELL Technologies

Whitepaper

Achieving High-Density 5G O-RAN Emulation

A Blueprint for Carrier-Grade Virtualization on Dell PowerEdge Servers, powered by AMD EPYC Gen5

SIMNOVUS

Table of contents

Executive Summary

Telecommunications providers are rapidly adopting Open Radio Access Networks (O-RAN). This shift has created a growing need for high-density, highly reliable testing platforms that go beyond what standard enterprise virtualization can deliver.

Simnovus and Dell have engineered a blueprint for achieving zero-drop, 25G fronthaul O-RAN 7.2 split emulation. The approach works by closely aligning Data Plane Development Kit (DPDK) workloads with the physical architecture of modern silicon.

This whitepaper outlines how we transformed a commercial off-the-shelf (COTS) Dell PowerEdge R7725 server into a high-performance O-RAN emulation environment. The server was equipped with two AMD EPYC Gen5 128-core CPUs and 1TB of DDR5 RAM, running Ubuntu Server 24.04.4 LTS. The result: a fully virtualized platform capable of sustaining 4.1 GHz compute across 240 perfectly isolated cores.

1. The Virtualized Edge Challenge

Validating O-RAN at scale requires testing platforms that can simultaneously emulate both Radio Units (RUs) and User Equipment (UEs). These platforms must also adhere to strict nanosecond-level timing windows.

Traditional Network Functions Virtualization (NFV) approaches often introduce hypervisor jitter, cache thrashing, and memory bandwidth bottlenecks. These issues result in dropped or delayed fronthaul frames.

To address these physical constraints, our solution relies on three key components:

- KVM/Libvirt virtualization
- Strict SR-IOV PCIe passthrough to bypass software virtual switches (vSwitches)
- KVM PTP hardware clock synchronization to maintain a tight, approximately 56-nanosecond RMS offset across multiple guest virtual machines (VMs)

Figure 1: Dell PowerEdge R7725

2. Navigating the Power-State Trap

The first major challenge in ultra-low latency computing is the conflict between traditional DPDK tuning methods and modern processor power management.

Historically, engineers pinned cores for DPDK polling loops by passing processor.max_cstate=0 and idle=poll to the kernel to prevent CPU sleep states. However, on modern AMD Zen architectures, the System Management Unit (SMU) requires C-state visibility to calculate thermal budgets. Masking these states forces the SMU to choose the safest frequency to avoid thermal saturation across all cores (1.2 GHz in our case)

The Solution: We removed the legacy kernel parameters and configured the amd_pstate=passive CPU frequency governor instead. This allowed the host operating system to give AMD's Precision Boost the thermal data it needed. The result was a sustained, unthrottled 4.1 GHz across all isolated virtual CPUs, delivering the raw compute required for high-speed shared memory polling.

3. L3 Cache Integrity and the 8-Core CCX Rule

Raw compute power is insufficient if the memory access pipeline is highly fragmented. Initial topologies mapped DPDK workloads sequentially, inadvertently spanning the physical Infinity Fabric boundaries of the AMD processor.

On EPYC Gen5 processors, the L3 cache boundary is fixed at 8 cores per Core Complex (CCX). When a workload crosses multiple CCXs, threads must communicate across the Data Fabric. This introduces small delays, known as micro-stalls, that cause missed transmission windows.

The Solution: Simnovus implemented an aggressive, mathematically precise NUMA and CCX pinning strategy:

1. **Housekeeping Isolation:** Guest OS background tasks and hypervisor emulator threads were physically banished to the host's non-isolated housekeeping cores.

2. **Workload Binning:** The O-RAN emulation pipelines were decoupled. Downlink/PRACH threads were binned into one pristine 8-core CCX, while Uplink threads were binned into an adjacent CCX.

By mapping the logical 5G data planes directly onto the physical topology of the silicon, workloads execute out of dedicated, uncontested physical L3 caches.

4. Shattering the Data Fabric Ceiling

Even with perfect cache isolation, scaling to hundreds of active virtual cores created heavy shared memory traffic between the DPDK and the user-space application workloads. Hardware profiling using AMD μProf revealed that the system's Data Fabric bandwidth was flatlining at approximately 307 GB/s.

While DDR5-6400 memory channels can theoretically deliver more than 1 TB/s, millions of microscopic, fragmented, cross-CCX read/write operations saturated the transactional capacity of the memory controllers. This is a clear example of how real-world performance can differ significantly from theoretical limits.

The Solution: With the hardware operating at its physical limits, optimization shifted to the environment and application level, using three targeted changes.

- **NPS4 Geometry:** The BIOS memory interleaving was configured to NUMA Per Socket 4 (NPS4), grouping memory channels into quadrants to keep memory fetches physically localized to specific CCXs.

- **DPDK EAL Optimization:** Launch parameters were updated to explicitly distribute buffers across all available silicon pathways (-n 12).

- **Burst Scaling:** DPDK polling loops were optimized to utilize rte_pause() during empty cycles and handle larger packet burst sizes (MAX_PKT_BURST), drastically reducing the transaction overhead on the Infinity Fabric.

5. Securing Nanosecond Determinism: Hardware PTP and Virtual Clock Synchronization

The Synchronization Mandate

In an O-RAN 7.2 split architecture, raw compute speed means nothing without extreme temporal precision. The transmission and reception of Control Plane (C-Plane) and User Plane (U-Plane) messages are governed by strict time boundaries. If a virtualized Radio Unit (RU) or UE emulator drifts by even a few microseconds, it will miss the strict OFDM symbol transmission windows. The result: discarded fronthaul frames and simulated radio link failures.

Traditional virtualized timekeeping—where a guest OS relies on standard Network Time Protocol (NTP) or emulated hypervisor timer ticks—introduces significant jitter. When hundreds of virtual cores are aggressively polling shared memory at 4.1 GHz, standard virtual clocks drift wildly under the interrupt load. To solve this, we engineered a direct, hardware-backed precision timing pipeline from the physical network interface straight into the guest operating system.

Establishing the Host Baseline

The timing architecture starts at the host's physical network layer. The virtualization host receives Precision Time Protocol (PTP, IEEE 1588v2) packets from a Grandmaster clock.

Using ptp4l, the host's physical NICs capture hardware-level timestamps as packets hit the silicon, completely bypassing the latency variability of the Linux network stack. The phc2sys daemon then disciplines the physical host's system clock to match the NIC's PTP Hardware Clock (PHC) with sub-microsecond accuracy.

Bridging the Hypervisor Gap: ptp_kvm

The key architectural challenge was delivering hardware-level Precision into multiple heavily isolated VMs without adding network overhead or hypervisor scheduling jitter.

Rather than forcing VMs to sync through a virtualized network bridge, the KVM hypervisor was configured to expose the host's disciplined clock directly to guests through the ptp_kvm kernel module. This paravirtualized driver creates a character device (/dev/ptp0) inside the guest OS. When the guest queries this device, it doesn't send a network packet. Instead, it reads the host's clock value almost instantly through a hypercall, effectively treating the host's system clock as a highly accurate, locally attached hardware oscillator.

Guest-Level Tuning with Chrony

Inside the guest VMs, the timekeeping architecture is optimized to consume this paravirtualized clock without interfering with DPDK workloads.

The chrony daemon is configured to ignore standard network time servers and rely exclusively on the KVM PTP device as its high-precision reference clock.

To keep clock synchronization stable under maximum load, the chronyd process inside the guest is pinned to the housekeeping virtual CPUs, specifically cores 0 and 1. This ensures time-discipline algorithms never run on the isolated CCXs dedicated to the DPDK RU and UE emulators. It also prevents cache thrashing and ensures the timekeeping thread is never starved by continuous polling loops.

The Result: Sub-100 Nanosecond Stability

By combining physical hardware timestamping, paravirtualized clock passthrough, and strict core-pinning isolation, the timekeeping architecture remains stable even under the massive compute load generated by the O-RAN data plane.

Telemetry captured at full saturation, with 240 cores pulling approximately 307 GB/s of Data Fabric bandwidth, showed that chrony within the guest VMs maintained a stable, approximately 56-nanosecond RMS offset from the host clock. This confirms that high-density, fully virtualized testbeds can meet the rigorous, nanosecond-level synchronization requirements of carrier-grade O-RAN deployments.

Conclusion

The successful deployment of a high-density Dell COTS platform marks a meaningful shift in 5G infrastructure testing. By aligning software design with the physical realities of modern silicon, from C-state thermal management to 8-core L3 cache boundaries and Data Fabric transactional limits, we've shown that carrier-grade density and strict O-RAN 7.2 determinism can be achieved simultaneously on COTS hardware.

Click here for additional information on Simnovus.

Learn more about Dell solutions

Contact a Dell Technologies Expert

View more resources

Join the conversation with #ORAN