



Contribution ID: 239

Type: **not specified**

Optimizing traffic receive (RX) path in Linux kernel MANA Driver for larger PAGE_SIZE systems.

Azure VMs have historically and popularly utilized the x86-64 architecture, which operates with a standard 4KB base page size. However, with the recent onboarding of ARM64-based VMs (ex. GB200), we now leverage an architecture that offers three configurable base page size options within the Linux kernel (4KB, 16KB, 64KB). This architectural flexibility, while beneficial in many use cases like model training, has exposed a significant inefficiency in the Mana driver's RX buffer handling. The current implementation allocates RX buffers on a per-page basis, which is suboptimal when a larger base page size, such as 64KB, is configured. This results in a single 1500-byte MTU packet consuming an entire 64KB page, leading to significant memory inefficiency and reduced network performance.

This talk will present a recent Linux kernel patch that addresses this issue. The optimization introduces the use of page_pool fragments and leverages pre-DMA mapped page pools for each RX queue. This approach allows the driver to efficiently utilize a single page for multiple network packets, providing substantial benefits across both architectures. For ARM64-based VMs configured with 64KB pages, this optimization reduces memory overhead per packet by over 97% and yields a performance improvement in throughput of approximately 200%. The same patch also provides a significant benefit to x86-64 VMs, with a reduction in memory overhead of around 50% and a performance improvement of approximately 15%. We will discuss the details of this optimization and its impact on performance across Azure cloud environment.

Primary authors: Mr ROY, Dipayan (Microsoft); ERNI, Sri Satya Vennela

Presenters: Mr ROY, Dipayan (Microsoft); ERNI, Sri Satya Vennela

Session Classification: Networking Track

Track Classification: Networking Track