

Cnuas Peer Memory, Datasheet



DMA-BUF interoperability between CnuasGPU memory and CnuasNIC RDMA

Document DS-CNU-023 • **Revision** A • **Issued** 23 August 2026 • **Status** Preview, v0.1 implemented; guest end-to-end validation gated

Item	Value
Part	libcnuaspeermem
Type	GPU-memory export and RDMA registration library
Version	0.1.0
Status	Implemented and host-tested; live guest transfer validation is gated
Repositories	PacketFive/CnuasGPU, PacketFive/CnuasNIC

Functional scope

Cnuas peer memory exposes CnuasGPU memory to the software CnuasNIC through the standard Linux DMA-BUF RDMA interface. It does not claim physical PCIe peer-to-peer bandwidth, latency, or direct hardware DMA.

1. Overview

Cnuas peer memory lets a CnuasNIC memory region name an allocation in a CnuasGPU device-memory window without a host staging buffer in the application's data flow. `cnuasgpu.ko` exports an owned, page-aligned allocation or subrange as a DMA-BUF. `libcnuaspeermem` passes that descriptor to standard `ibv_reg_dmabuf_mr()`, and `cnuas_ib.ko` retains an I/O-safe mapping for the memory-region lifetime.

This path is separate from CnuasLink. Peer memory connects GPU memory to the RDMA network through CnuasNIC; CnuasLink carries framed GPU-to-GPU traffic over the accelerator fabric.

2. Data path

Diagram available in the online documentation at <https://github.com/PacketFive/cnuas>.

1. Allocate page-aligned CnuasGPU memory.
2. Export the allocation or an owned subrange as a DMA-BUF.
3. Register the descriptor with a CnuasNIC protection domain.
4. Use the resulting `lkey` or `rkey` in RDMA work requests.
5. Deregister the memory region and close the peer-memory region.

3. Functional specifications

Capability	State
Whole-allocation and page-aligned subrange export	Implemented
Read-only and close-on-exec export flags	Implemented
Export lifetime after the originating GPU descriptor closes	Implemented
Standard <code>ibv_reg_dmabuf_mr()</code> registration	Implemented
SEND and RECV access through DMA-BUF-backed MRs	Implemented



Capability	State
RDMA READ and WRITE access through DMA-BUF-backed MRs	Implemented
Atomic access through DMA-BUF-backed MRs	Implemented with software serialization
Shared and static userspace libraries	Implemented
Host ABI, validation, ownership, and error-path tests	Passing
Live guest export-lifetime test	Gated on /dev/cnuasgpuN
Live CnuasGPU-to-CnuasNIC transfer test	Gated on GPU and RDMA devices

The implementation uses the upstream DMA-BUF verbs path. It does not implement the removed out-of-tree `ib_register_peer_memory_client()` API used by legacy peer-memory clients.

4. Userspace interface

Header: `cnuas_peermem.h`

Group	Interface
Capability discovery	<code>cnuas_peermem_have_verbs</code> , <code>cnuas_peermem_have_dmabuf_mr</code> , <code>cnuas_peermem_query_caps</code>
Export	<code>cnuas_peermem_export</code>
Registration	<code>cnuas_peermem_register</code> , <code>cnuas_peermem_unregister</code>
RDMA keys	<code>cnuas_peermem_lkey</code> , <code>cnuas_peermem_rkey</code> , <code>cnuas_peermem_mr</code>
Region properties	<code>cnuas_peermem_fd</code> , <code>cnuas_peermem_offset</code> , <code>cnuas_peermem_size</code> , <code>cnuas_peermem_flags</code>
Lifetime	<code>cnuas_peermem_close</code>
Diagnostics	<code>cnuas_peermem_build_info</code>

Functions return zero or a negative `errno` and do not abort or log. The caller retains ownership of the GPU descriptor and protection domain. A peer-memory region owns only the DMA-BUF descriptor and verbs memory region that it creates.

5. Kernel interfaces

ComponentInterface	Purpose
CnuasGPU <code>CNUASGPU_IOC_CAPS</code>	Report export support, alignment, and limits
CnuasGPU <code>CNUASGPU_IOC_EXPORT_DMABUF</code>	Export an owned allocation range
CnuasNIC <code>ib_device_ops.reg_user_mr_dmabuf</code>	Import a DMA-BUF as an RDMA memory region
rdma-core <code>ibv_reg_dmabuf_mr()</code>	Standard userspace registration operation

The GPU exporter keeps exported allocations and device state alive until the last DMA-BUF reference closes. The NIC importer uses `iosys_map` accessors so a device BAR is not treated as ordinary host memory. Memory regions are reference-counted across posted receives and in-flight READ or atomic work.



6. Build and validation

```
make -C src/cnuasgpu/peermem
make -C src/cnuasgpu/peermem check
make -C src/cnuasgpu/peermem check-device
```

Test	Environment	Coverage
peermem_abi_smoke	Any build host	ABI layout, ioctl values, flags, alignment, bounds, overflow, reserved fields
peermem_api_smoke	Any build host	Ownership, NULL handling, errno behavior, invalid descriptors
peermem_lifetime_smoke	Guest with CnuasGPU	Export lifetime, busy allocation, subranges, orphan reclaim
cnuas-peermem-smoke	Guest with CnuasGPU and CnuasNIC	Export, DMA-BUF MR registration, RC transfer, byte comparison

Device-dependent tests return exit status 77 when their required interfaces are absent. A skip confirms the gate behaved correctly; it does not validate the live transfer path.

7. Integration information

Requirement	Value
GPU device	/dev/cnuasgpuN; /dev/cnuasgpu_hostN can export pinned host pages through the same ABI
GPU driver	cnuasgpu.ko or cnuasgpu_host.ko with DMA-BUF enabled
RDMA driver	cnuas_ib.ko with reg_user_mr_dmabuf
Userspace RDMA	rdma-core exposing ibv_reg_dmabuf_mr()
Build without verbs	Export and capability APIs remain available; registration returns unsupported
Detailed design	Cnuas Peer Memory
Validation status	Validation Matrix

8. Current limits

- Full live guest transfer validation remains gated.
- The software RNIC copies payloads with the CPU; this is not physical peer-to-peer DMA.
- Peer-memory atomics are serialized within CnuasNIC and cannot serialize against independent GPU-side mutation.
- Legacy nvidia-peermem client API compatibility is not provided.

9. Revision history

Revision	Notes
A	First publication. Records the implemented DMA-BUF exporter, userspace library, CnuasNIC importer, host tests, and gated live-device tests.

PacketFive, Packet Five Networks Ltd., Dublin, Ireland.

Cnuas Virtual AI/HPC Infrastructure is published at <https://github.com/PacketFive/cnuas> under the Apache License 2.0; read it at <https://github.com/PacketFive/cnuas/blob/main/LICENSE>.



Cnuas Virtual AI/HPC Infrastructure is emulation software. It is not affiliated with, endorsed by, or derived from any hardware vendor, and every device it models is a software artefact. The work is published by its authors in a personal capacity and is not sponsored or endorsed by any employer.

Specifications describe the referenced revision of the software and may change without notice. Contact info@packetfive.com.