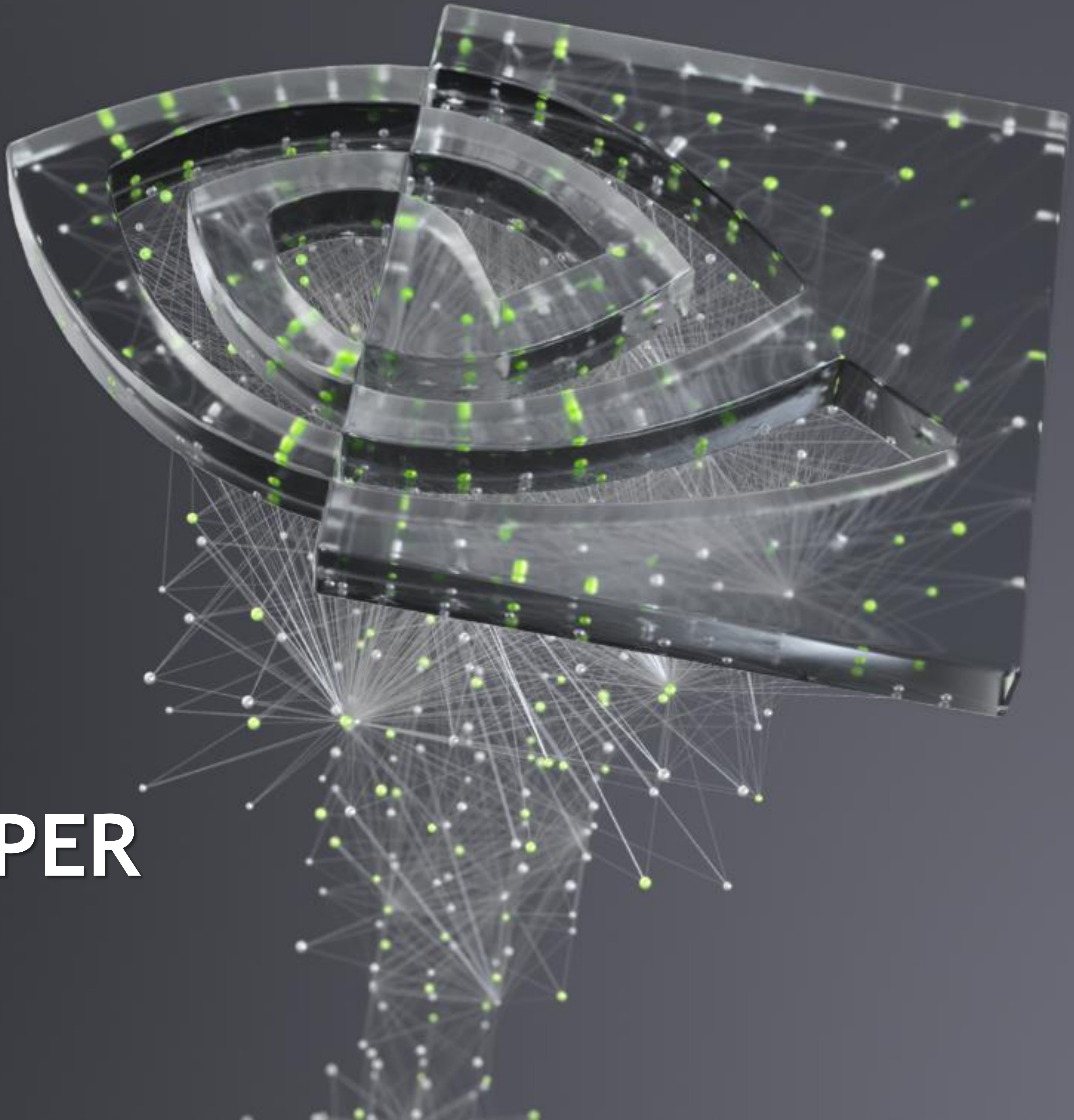
NVIDIA



UCX ON GRACE HOPPER

Akshay Venkatesh | UCF 2023

CONTRIBUTIONS

Akshay Venkatesh

Hessam Mirsadeghi

Jim Dinan

Nishank Chandawala

Sreeram Potluri

UCX core team

OUTLINE

UCX CUDA-awareness features

Grace-Hopper

Unified memory

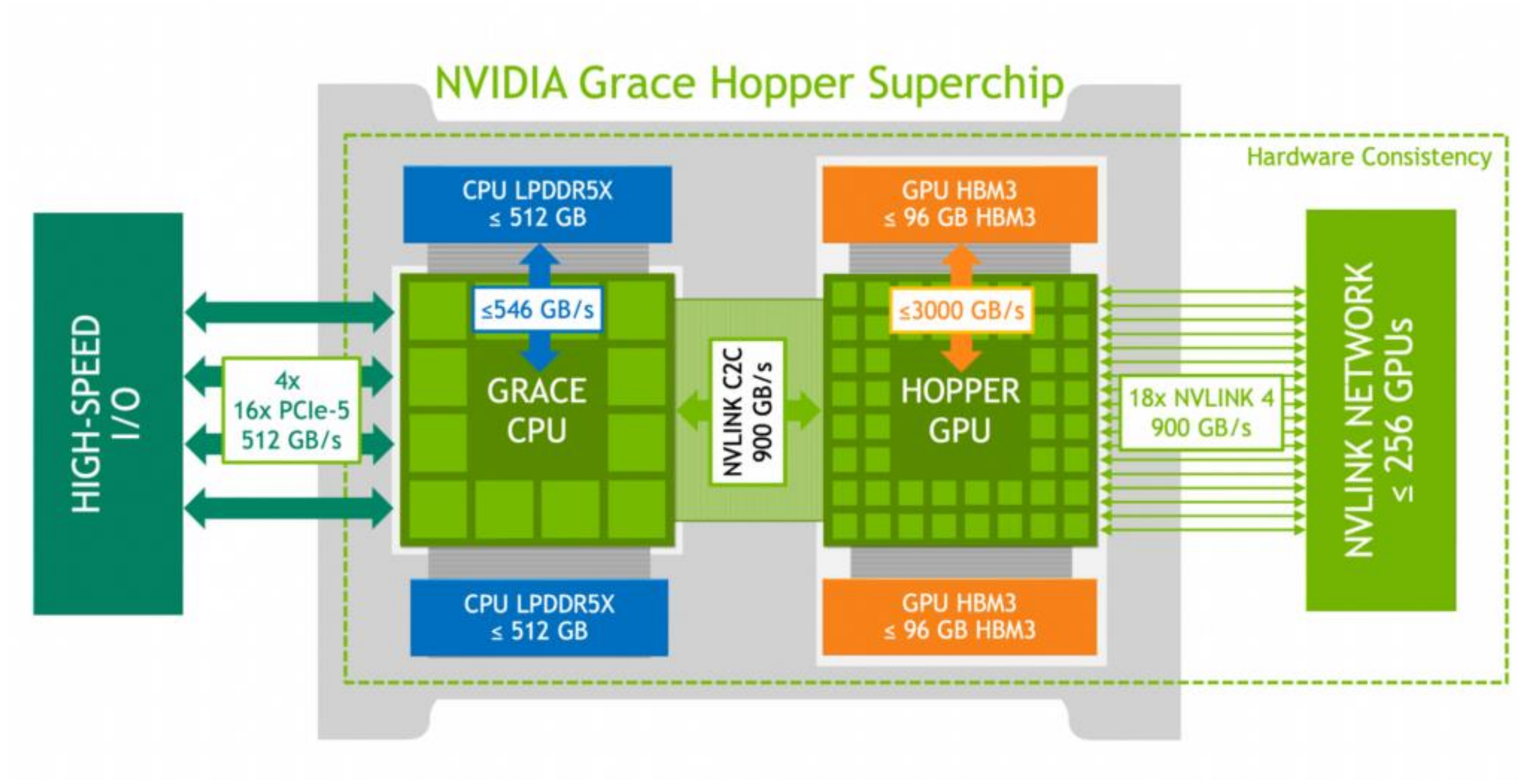
NVSwitch systems

Extended GPU memory

UCX FEATURES FOR GPU-AWARENESS

Feature	Description	Layer
IPC transport	Takes advantage of remote GPU mapping capability Takes advantage of NVLINK/high-bandwidth GPU-to-GPU fabric	UCT
Peer-to-peer transfers	Transports like IB allow GPU memory to be registered NIC can directly read/write from/to GPU memory Allows for zero-copy protocols Avoids staging requirements saving latency and PCIe bottlenecks	UCT
Copy transports	Use DMA engine or host-mapping of local GPU memory Needed for staging in pipeline protocols, and eager protocols	UCT
Memory detection	Interception of GPU alloc/free calls Needed to select transport/protocol Needed to safely de-register memory in the context of zcopy transports	UCM
Topology awareness	Distance between system devices are used to make protocol decisions in the form of bandwidth and latency estimations	UCS
UCP protocols v2	Memory type/location based selection of optimal protocols Without this, use of common rendezvous thresholds lead to poor perf	UCP

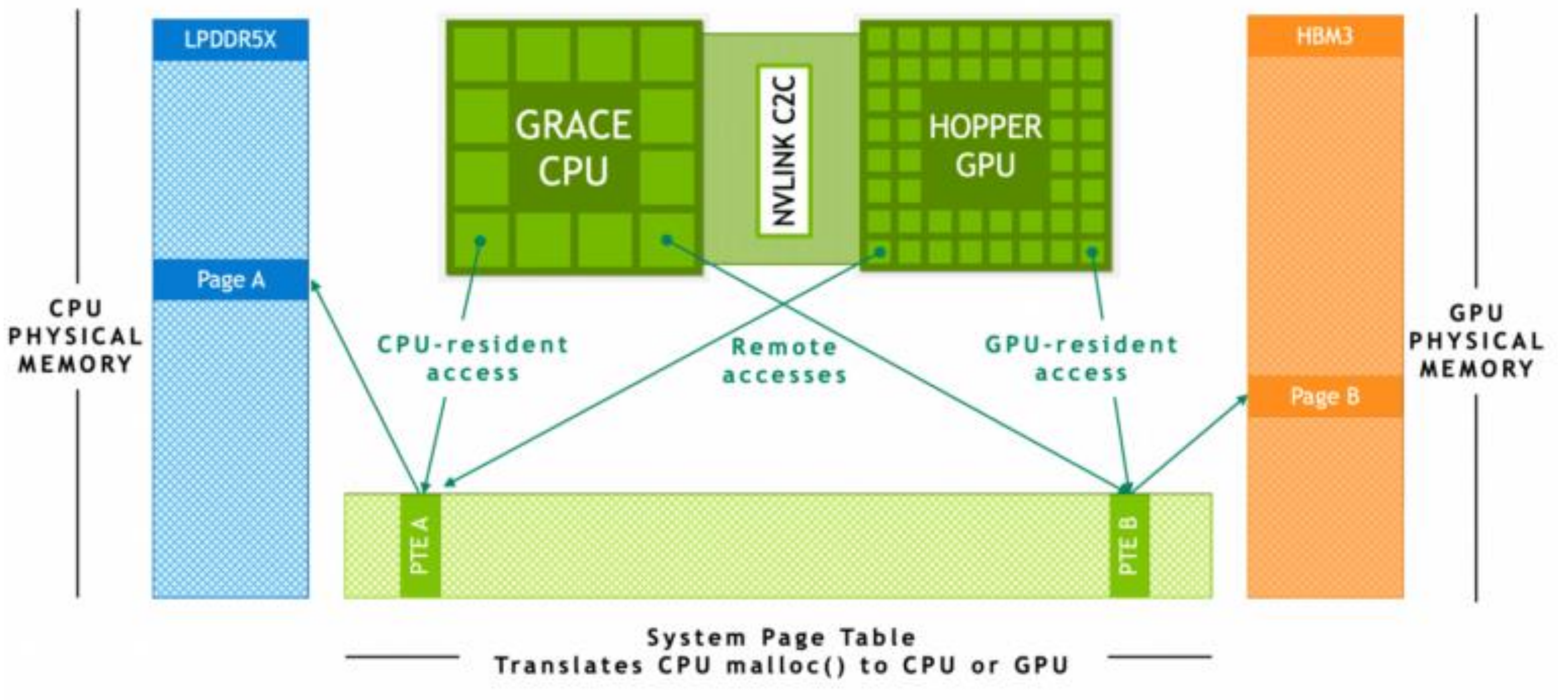
GRACE HOPPER SUPERCHIP AT A GLANCE



Feature	Description
Grace CPU cores (number)	Up to 72 cores
CPU LPDDR5X bandwidth (GB/s)	Up to 500GB/s
GPU HBM bandwidth (GB/s)	4TB/s HBM3 4.9TB/s HBM3e
NVLink-C2C bandwidth (GB/s)	900GB/s total 450GB/s per dir
CPU LPDDR5X capacity (GB)	Up to 480GB
GPU HBM capacity (GB)	96GB HBM3 144GB HBM3e
PCIe Gen 5 Lanes	64x
NVLINK Switch System	Up to 256 GPUs

Refer https://developer.nvidia.com/blog/nvidia-grace-hopper-superchip-architecture-in-depth

PROGRAMMING MODEL ON GRACE-HOPPER



ATS enables the CPU and GPU to share a single per-process page table

All CPU and GPU threads can access all system-allocated memory, which can reside on physical CPU or GPU memory

NVIDIA NVLink-C2C hardware-coherency enables the Grace CPU to cache GPU memory at cache-line granularity and for the GPU and CPU to access each other’s memory without page-migrations

IMPLICATIONS OF GLOBALLY-ACCESSIBLE MEMORY ON UCX

UCX pins memory prior to rendezvous transfers (used for large message range within/across nodes)

- Pinning will prevent appropriate page migrations on Grace Hopper if workload dictates such patterns
- Pinning causes remote access within the node and can reduce overall performance
- Pinning is performed by default by IB, CMA transports

Solution

- Detect Grace-Hopper platform
- Use On-Demand Paging (ODP) registration scheme on user memory of type sysmem, cuda-managed
- ODP allows pages to migrate

Caveats

- Does not solve intra-node problem
- Need to avoid ODP for UCX allocated internal memory as these are unlikely to be accessed by GPU threads

STATUS

Introduction of migratable memory types (https://github.com/openucx/ucx/pull/9370)

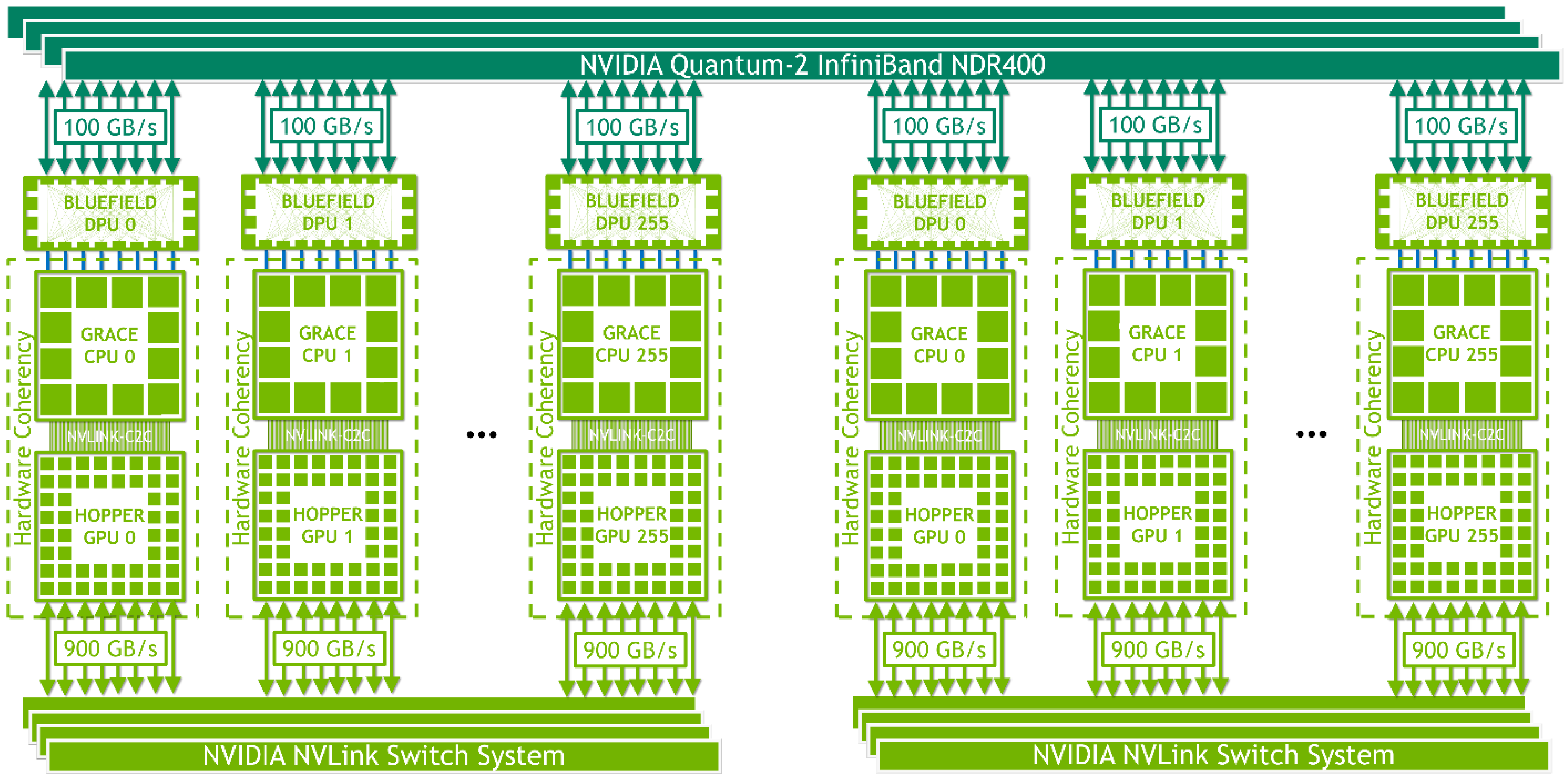
- Memory domain populates a bit array of memory types to be treated as migratable
- Transports capable of on-demand/non-blocking registration then use non-pinning registration for such memory

Extensions to ucp_mem_map API (PR pending)

- To allow users to explicitly request pinning or blocking registration of memory regions to avoid ODP registration

Need to explore prefetching policies and its effects on page migration

HGX GRACE HOPPER SYSTEM OVERVIEW



<= 256 GPUs in NVLINK-connected system; GPU threads in NVSwitch-connected domain can address 150 TB memory

NVIDIA ConnectX-7 NICs or BlueField-3 DPUs paired with Quantum 2 NDR switches or other OEM-defined I/O provides additional networking

IMPLICATIONS OF MULTI-NODE NVLINK SYSTEM ON UCX

Import/export mapping API handle different from API associated with cudaMalloc

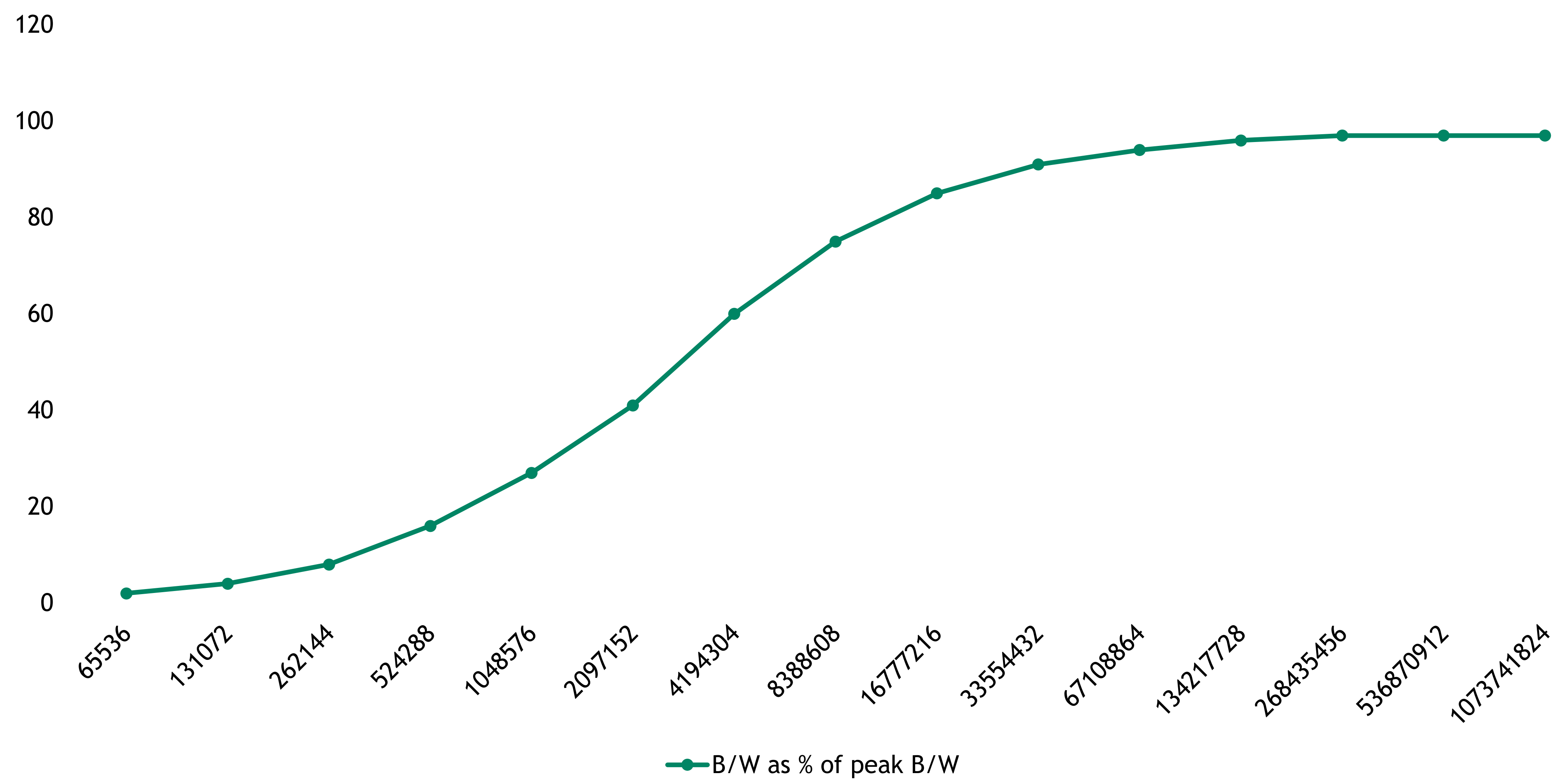
IPC transports like CUDA_IPC considered shared memory transport (DEVICE_TYPE_SHM)

Prevents IPC transport's implementation from being used for multi-node nvlink capabilities

Changing IPC transport to DEVICE_TYPE_NET requires changes to

- Connection-establishment logic
- Device reachability logic
- Keepalive feature logic

INTERNAL IMPLEMENTATION ACROSS 2-NODES (WIP)

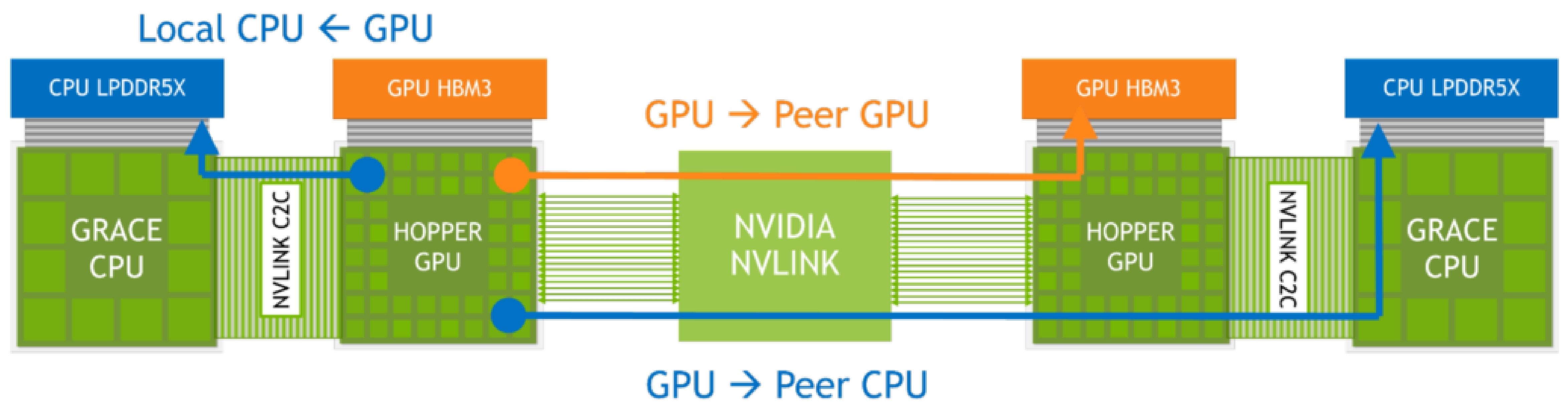


Ported CUDA_IPC transport to use new Import/export API needed to use multi-node nvlink capabilities

Discussing NVML-based reachability logic to ensure GPUs outside NVLINK domain (for systems with > 256 GPUs) avoid use of IPC transport

Relaxing some keepalive requirements for a workaround but discussing long-term solution

EXTENDED GPU MEMORY (EGM)



Grace Hopper allows new allocations on system memory as extensions of GPU memory

EGM over NVLINK-C2C enables all GPUs to access <= 150 TB of memory in multi-node NVSwitch-connected system

GPU threads can access at min(NVLINK-C2C BW, GPU NVLINK BW)

IMPLICATIONS OF EGM ON UCX

Import/export API similar to multi-node nvlink IPC to be used for EGM

CUDA_IPC transport needs to be extended to support cuda allocations backed by system memory

=> CUDA_IPC ‘network’ device avails itself for system memory inter-node transfers

UCP should fallback to using high-bandwidth transports like IB if remote system memory is not accessible through IPC

=> rkey_unpack failure should result in UCP protov2 to gracefully fallback to next best transport for the operation

The same fallback logic would apply to cases where remote GPU memory is not addressable due to other limitations

OTHER REQUESTS

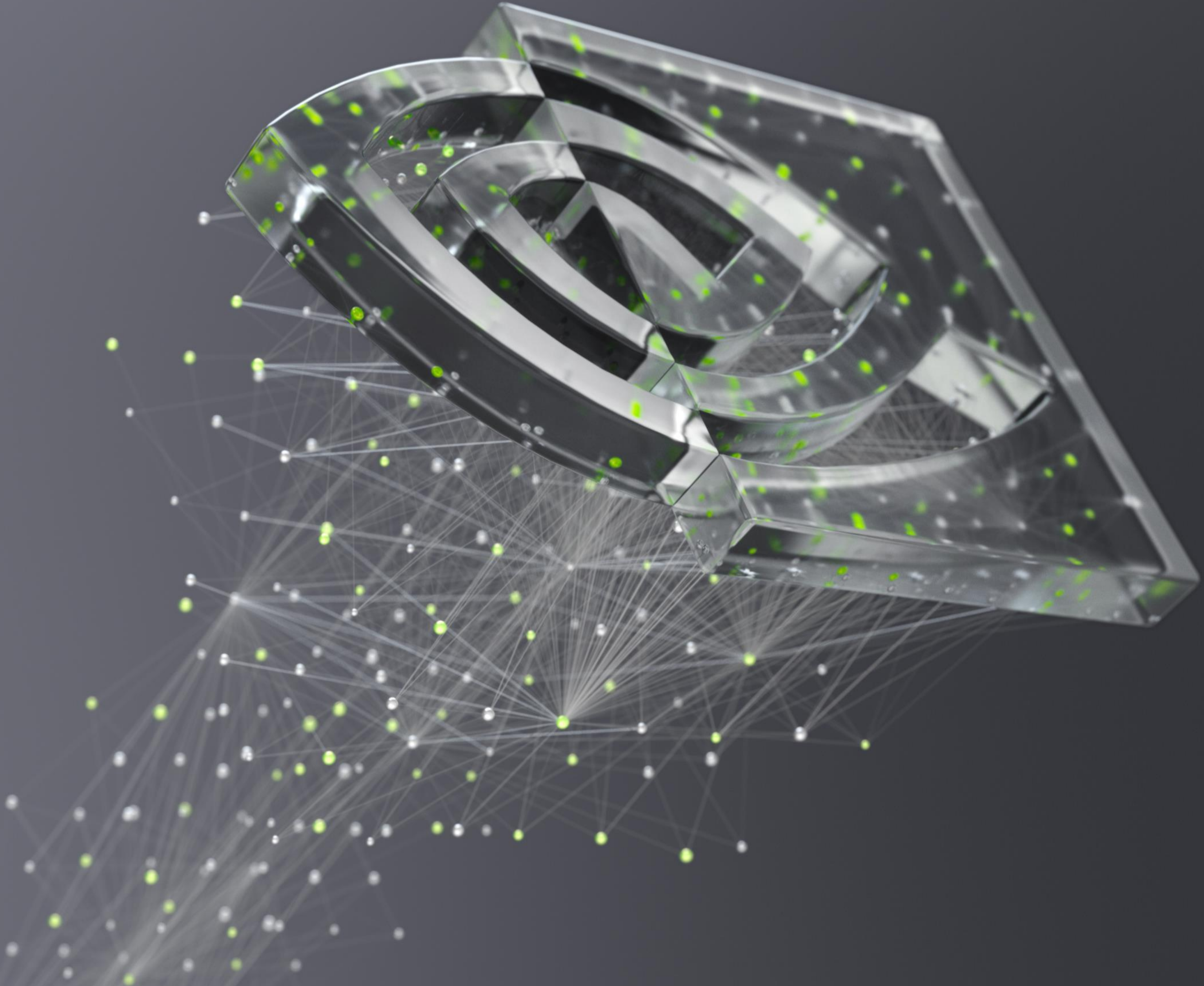
Backing page location exposure

- CUDA to soon introduce API to query physical locations backing user pages
- When pipeline protocols are chosen, bounce buffers should be closer to pages
- => Need protocols v2 to factor use of device bounce buffers in protocols
- => Need pipeline protocols that can use device bounce buffers and EGM bounce buffers

May need to avoid use of CMA for intra-node transfers between system memory

- To avoid pinning
- To get better bandwidth (CNVLINK has 100 GB/s peak b/w vs C2C->NVLINK->C2C has 150 GB/s b/w)
- => Need pipeline protocols that use device bounce buffers or EGM bounce buffers

Prefetching policies for migratable memory with ODP



NVIDIA®