Hewlett Packard
Enterprise

Designing GPU-aware OpenSHMEM for HPE Cray EX and XD

Naveen Namashivayam Ravichandrasekaran,
Nathan Wichmann,
Danielle Sikich, Elliot Rohaghan,
Md Rahman, William Okuno

OpenSHMEM Programming Model Review

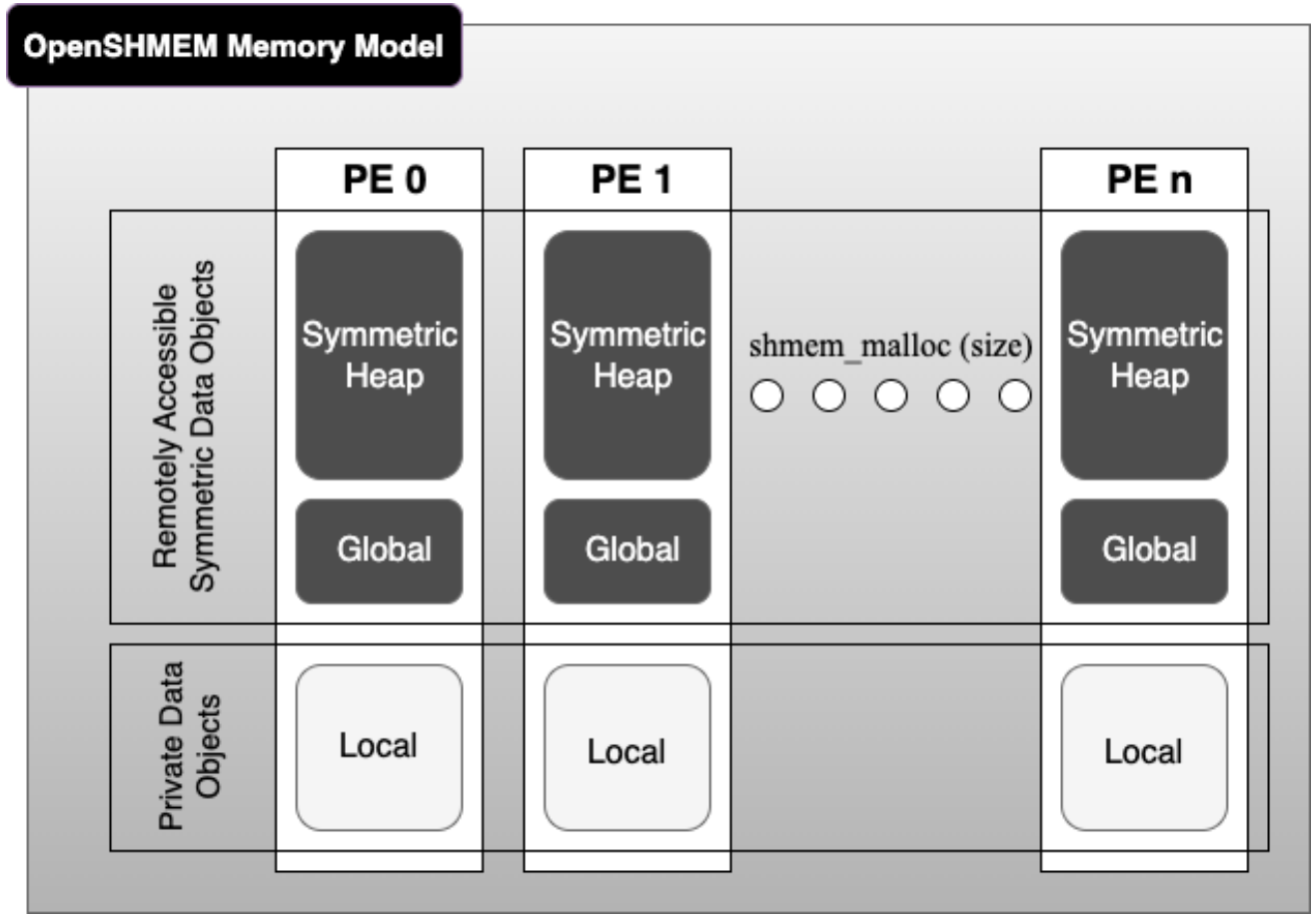
- OpenSHMEM is a Partitioned Global Address Space (PGAS) model
  - Enabling global memory spaces partitioned across PEs

- Data accessed asynchronously, in a one-sided manner, using Put, Get, and Atomic operations
  - Blocking and non-blocking APIs available
  - Thread-hot support via context
  - Various collectives also supported

- Memory ordering semantics provided via shmem_quiet, shmem_barrier and more

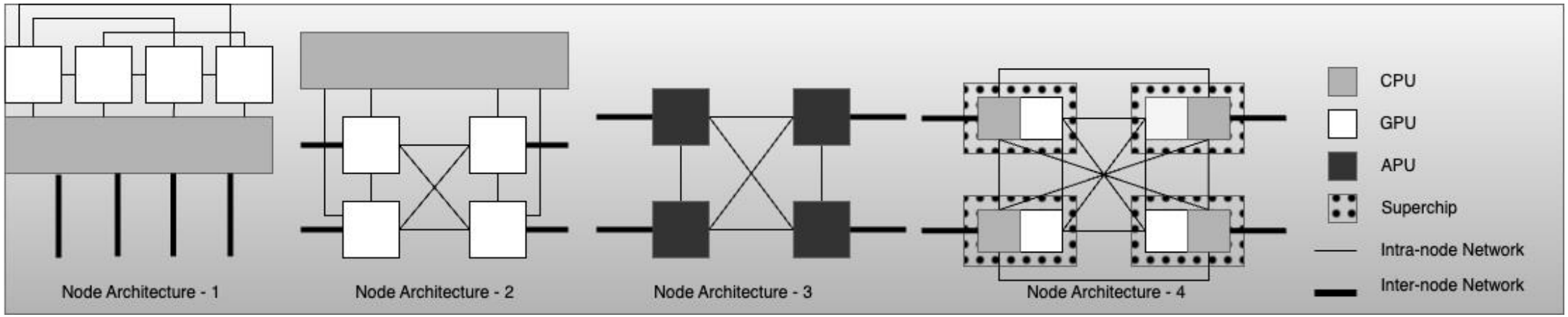
```
shmem_put( &dst[i], &src[i], nelems, target_PE );

shmem_put_nbi( &dst[i], &src[i], nelems, target_PE );

shmem_ctx_put_nbi( ctx, &dst[i], &src[i], nelems, target_PE );

shmem_atomic_fetch_add_nbi(&ret,&target[0],1,target_pe);

shmem_quiet();
```

OpenSHMEM Memory Management and Layout



- Communication executed on Symmetric Data Objects (SDOs)
- Long had only two SDO spaces:
  - Symmetric heap (default)
    - –Allocated using shmem_malloc (size) etc.
    - –Collectively allocated across all PEs
  - Global / Static variables
- Private Data Objects
  - Local to each process
    - –Normal heap and stack
    - –No symmetric requirements
  - Not accessible via SHMEM API

Heterogeneous Compute Node Architectures



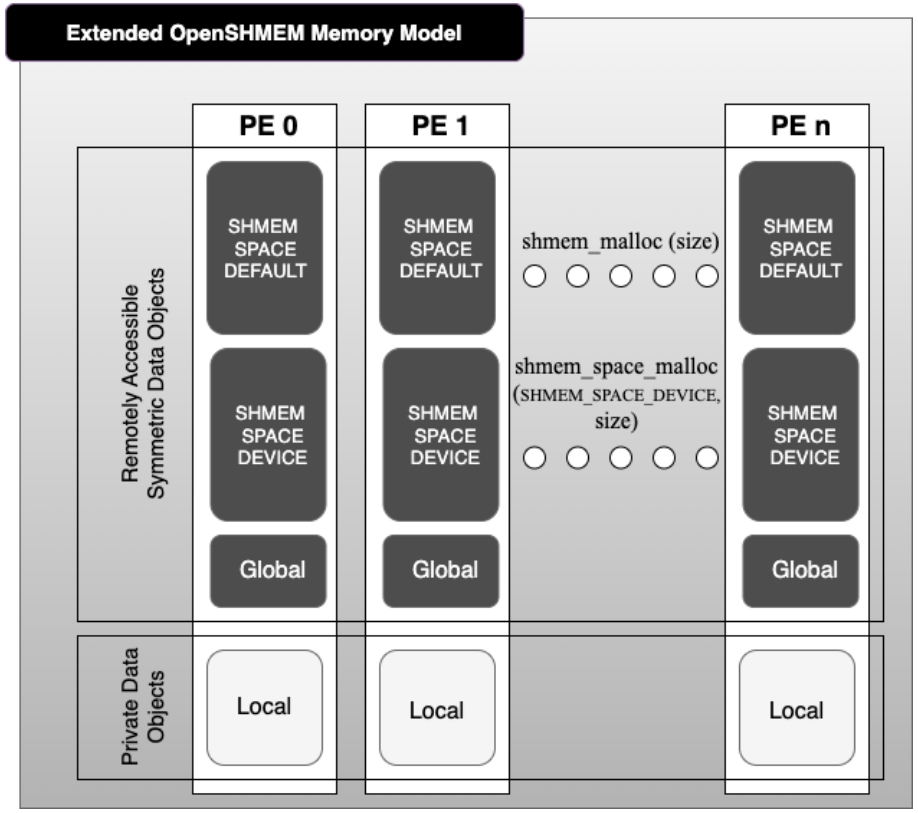
- Heterogeneous Compute Nodes are common in HPC and AI systems
  - 9 of the top 10 system on Top500

- Differ significant across multiple dimensions
  - Discrete accelerators
  - APUs and APU-like
  - CPU – GPU ratio
  - NIC connectivity: Number and location

- Common theme of addition memory domains, either physical or programming model

- OpenSHMEM currently lacks visibility into memory placement and accessibility

OpenSHMEM Memory Management and Layout



- Introducing new spaces into SHMEM memory model
- Default space is the same as the old symmetric heap
- Device space specifically allocated on and accessible by the accelerator
- SHMEM communication APIs can now use variables in either space without any change to the API

```
int *var_a = shmem_space_malloc(SHMEM_SPACE_DEFAULT,size*nelems);
int *var_b = shmem_space_malloc(SHMEM_SPACE_DEVICE,size*nelems);
…
shmem_put_nbi( &var_b[i], &var_a[i], nelems, target_PE );
```

Integer Sort Prototype

- Integer Sort on the GPU interleaved with a global data exchange
  - Critical to large-scale data processing
  - Prototype code name A47

- Series of GPU-kernels implement a bucket-sort into a small buffer
  - Count the number of elements that will be sent to each PE
    - Heavy use of GPU-atomics to GPU-shared memory
  - Pointer calculations to compute starting location and reserve space in the buffer
  - Sort data from the table into the buffer
    - Heavy use of GPU-atomics to GPU-shared memory

- Global exchange to move the data from the local buffer to the destination PE
  - All-to-All-V type of exchange, but written using one-sided SHMEM calls
  - Nearly-in-place global sort

- Executed on HPE Cray EX across 128 MI300A GPUs

A47: 8-byte, Single Word Elements

```
*****************************************************
* A47
* NPES                                = 128
* Est Num Elements per PE             = 5261334937
* sizeof(ELEMENT_TYPE) Word           = 8
* Words per Element                   = 1
* Headroom                            = 2.00   %
* OMP Threads                         = 24
* Rand Seed                           = 1746130763
* table array size/PE (MiB)           = 40960.000
* Scratch Pad size/PE (MiB)           = 2000.000
* Est Memory footprint/PE (MiB)       = 85920.000

Test      A_MsizeB  TotTime SortTime  CpyTime ExchTime LB_GiB/s/PE UB_GiB/s/PE  Check
HA_HI     16384000   29.950    1.547    7.377   19.082        13.1        20.5  PASSED

A47 has sorted things out!
*****************************************************
```

- Run on 128 GPUs
  - 1 PE per GPU
- This run used 24 OpenMP threads
  - Mostly to initiate PUTs
- HA_HI mean Host Allocated Host Initiated
  - I.E., not GPU-Aware SHMEM
- Exchange Bandwidth
  - LB= Lower Bound= Total_data/TotTime/NPES
  - UB= Upper Bound= Total_data/ExchTime /NPES
- UB very near Peak Bandwidth for MI300A with SS-200

A47: 8-byte, Single Word Elements

```
**************************************************
* A47
* NPES                              = 128
* Est Num Elements per PE           = 5261334937
* sizeof(ELEMENT_TYPE) Word         = 8
* Words per Element                 = 1
* Headroom                          = 2.00   %
* OMP Threads                       = 24
* Rand Seed                         = 1746130763
* table array size/PE (MiB)         = 40960.000
* Scratch Pad size/PE (MiB)         = 2000.000
* Est Memory footprint/PE (MiB)     = 85920.000

Test      A_MsizeB  TotTime  SortTime   CpyTime  ExchTime LB_GiB/s/PE UB_GiB/s/PE  Check
HA_HI     16384000   29.950     1.547     7.377    19.082         13.1         20.5  PASSED

A47 has sorted things out!
**************************************************
```

- 2 copies of the table must be held in memory at the same time
- TotTime = Total time to execute the HA_HI call
- SortTime = Time to do the local sort, pre-exchange, on the GPU
- CpyTime = Time to copy data back to host
  - Slower than expected
  - Taking up about 25% of the time
- Exchange Time = A2A time, including FADDs

A47: 8-byte, Single Word Elements

```
**************************************************
* A47
* NPES                              = 128
* Est Num Elements per PE           = 12756052869
* sizeof(ELEMENT_TYPE) Word         = 8
* Words per Element                 = 1
* Headroom                          = 1.00   %
* OMP Threads                       = 24
* Rand Seed                         = 1746130326
* table array size/PE (MiB)         = 98304.000
* Scratch Pad size/PE (MiB)         = 1000.000
* Est Memory footprint/PE (MiB)     = 99304.000

Test       A_MsizeB  TotTime SortTime  CpyTime ExchTime LB_GiB/s/PE UB_GiB/s/PE  Check
GA_HI       8192000   53.085    4.144    0.000   48.346        17.9        19.7  PASSED

A47 has sorted things out!
**************************************************
```

- GA_HI mean GPU Allocated Host Initiated
  - I.E., GPU-Aware SHMEM
- CpyTime is eliminated
- 96 GiB table
- Exchange still achieves excellent bandwidth
  - LB BW approaching UB BW

A47: 8-byte, Single Word Elements

```
**************************************************
* A47
* NPES                              = 128
* Est Num Elements per PE           = 12756052869
* sizeof(ELEMENT_TYPE) Word         = 8
* Words per Element                 = 1
* Headroom                          = 1.00   %
* OMP Threads                       = 24
* Rand Seed                         = 1746130326
* table array size/PE (MiB)         = 98304.000
* Scratch Pad size/PE (MiB)         = 1000.000
* Est Memory footprint/PE (MiB)     = 99304.000
```

Test	A_MsizeB	TotTime	SortTime	CpyTime	ExchTime	LB_GiB/s/PE	UB_GiB/s/PE	Check
GA_HI	8192000	53.085	4.144	0.000	48.346	17.9	19.7	PASSED
GA_HI_OO	4096000	47.295	0.922	0.000	45.637	20.1	20.8	PASSED

```
A47 has sorted things out!
**************************************************
```

- GA_HI_OO version uses OpenMP thread specialization
  - One thread launches sort kernels while rest exchange
- SortTime completely hidden under the ExchTime
- Exchange LB BW nearly the same as UB BW
  - About 10% faster in total

Conclusions

- Extending OpenSHMEM to support GPU-Aware allocation and communication empowers users to efficiently leverage heterogenous GPU platforms
  - Eliminates the need for data copies on all heterogenous node architectures
  - Allows for more efficient use of memory
  - Still achieves excellent performance
  - No scaling limitations

- Future work will
  - Extend this to investigate Kernel-Trigger, Stream-Triggered, and Kernel Initiated comms
  - Run on Slingshot-400 hardware

Thank you