CUDA Unified Memory

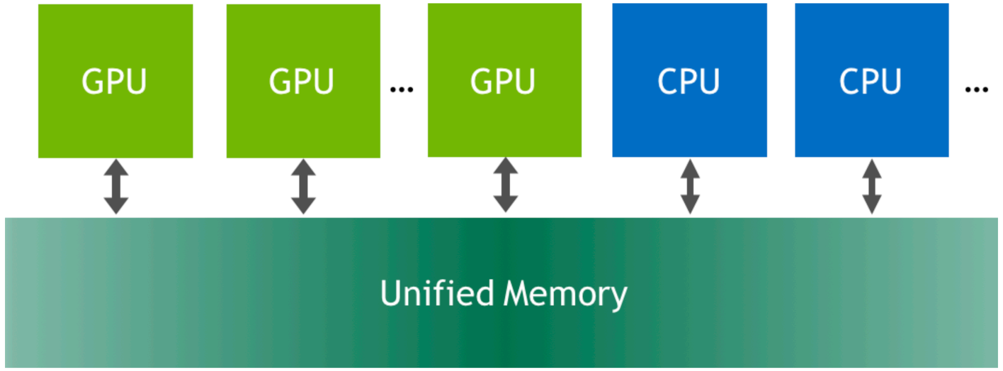
Stefano Markidis

Two Key-Points

1. CUDA Unified is a single memory space for host and device memories
2. We can eliminate to explicitly move data from CPU to GPU and vice-versa by using CUDA Unified Memory. CUDA runtime automatically takes of data migration

What is Unified Memory?

- Unified Memory is a single memory address space accessible from any processor in a system
  - allocate data that can be read or written from code running on either CPUs or GPUs.
  - the CUDA system software takes care of migrating memory pages to the memory of the accessing processor.



Allocating Unified Memory

- To allocate Unified Memory, we replace calls to `malloc()` or `new` with calls to `cudaMallocManaged()`
  - an allocation function that provides a pointer accessible both from CPU and GPU

```
cudaError_t cudaMallocManaged(void** ptr, size_t size);
```

Code Example

- `x` and `y` are accessible from both CPU and GPU
- No need for `CudaMemcpy()`
- CUDA driver takes care of the data movement automatically.
- Kernel launch is asynchronous with respect to the host
    - Need to **explicitly synchronize** on the host side **before directly accessing the output of the kernel**

```
int main(void)
{
  int N = 4096;
  float *x, *y;

  // Allocate Unified Memory -- accessible from CPU or GPU
  cudaMallocManaged(&x, N*sizeof(float));
  cudaMallocManaged(&y, N*sizeof(float));

  // initialize x and y arrays on the host
  for (int i = 0; i < N; i++) {
    x[i] = 1.0f; y[i] = 2.0f;
  }
  // Launch kernel on 4096 elements on the GPU
  int blockSize = 256;
  int numBlocks = (N + blockSize - 1) / blockSize;
  add<<<numBlocks, blockSize>>>(N, x, y);
  cudaDeviceSynchronize();
  ...
```

How Does it Work?

- CUDA Unified memory works differently depending on whether GPU is pre- or post Pascal generation.
- On pre-Pascal GPUs, `cudaMallocManaged()` allocates managed memory *on the GPU*.
  - Internally, the driver also sets up page table entries for all pages covered by the allocation
  - Upon launching a kernel, the CUDA runtime must migrate all pages previously migrated to host memory back to the GPU memory
- On post-Pascal GPUs, managed memory may not be physically allocated when `cudaMallocManaged()` returns
  - It may only be populated on access
  - Pages and page table entries may not be created until they are accessed by the GPU or the CPU

To Summarize

- CUDA Unified is a single memory space for host and device memories.
- We can eliminate to explicitly move data from CPU to GPU and vice-versa by using CUDA Unified Memory. CUDA runtime automatically takes of data migration