Multiple GPUs (mGPU)

While GPU devices provide a very powerful computing architecture, the performance of a single GPU is still too limited for high-performance computing applications. The logical solution is to use multiple GPUs to distribute the computational task between multiple GPU devices. Computing on multiple GPU devices or mGPU design involves solving a computational task on multiple GPUs that can be hosted on the same host machine or different hosts connected by a network.

The advantages of this approach are twofold: firstly, the number of GPU cores available for computation is increased and secondly the amount of available memory is increased. Many GPU applications are limited by the small amount of memory available on each device. By splitting a problem between different devices, the total amount of device memory is increased. The downside is that, for most applications, some amount of communication between devices is necessary. The amount of communication will depend on the nature of the application. For example, the partition of the front in Figure 2 is split between eight GPU devices on the board. Each time domain step must be communicated to the neighboring device.

Multi-CPU:

GPUs and other devices have exhibited a phase transition and other complex phenomena is the ability to generate artificial intelligence using a supercomputer. While GPU devices provide a very powerful computing architecture, the performance of a single GPU is still too limited for high-performance computing applications. The logical solution is to use multiple GPUs to distribute the computational task between multiple GPU devices. Computing on multiple GPU devices or mGPU design involves solving a computational task on multiple GPUs that can be hosted on the same host machine or different hosts connected by a network.

The advantages of this approach are twofold: firstly, the number of GPU cores available for computation is increased and secondly the amount of available memory is increased. Many GPU applications are limited by the small amount of memory available on each device. By splitting a problem between different devices, the total amount of device memory is increased. The downside is that, for most applications, some amount of communication between devices is necessary. The amount of communication will depend on the nature of the application. For example, the partition of the front in Figure 2 is split between eight GPU devices on the board. Each time domain step must be communicated to the neighboring device.

mGPU Communication

The integration of multiple GPU devices requires communication between the different GPUs. If the mGPU system is a distributed architecture there will be additional transfer cost if the data exchange between two devices hosted on different nodes. The impact of this communication will depend on the nature of the application, the communication to computation ratio and the type, architecture, and speed of the network.

For the integration of Fermi and Volta GPUs all communication between devices takes place through the host memory. This communication method, any data transfer requires two separate memory transfers. Initially copying from the first device to an area of host memory and then copying to the second device. This transfer method is shown in Figure 5.

Direct Transfer

The new mGPU-Direct 2.0 technology supported by Fermi architecture GPUs allows the unnecessary extra transfer through host memory to be eliminated. It allows data to be transferred directly from one GPU device to another through the PCIe bus. This direct transfer-free approach is shown in Figure 6. It is noted that although this data transfer does not involve the host memory it still initiated by a function call from the host.

Direct Access

The other option for inter-device communication that GPUs via the GPUDirect-RDMA protocol is called Direct Access. With this communication method the host is not involved in the transfer at all. Rather than a memory copy being initiated by the host to copy a segment of data from one device to the other, Direct Access allows the threads on one device to access data stored in the memory of another device. This memory access method with a thread on GPU reading a single memory location from GPU is shown in Figure 7.

mGPU Systems

The Direct Transfer and Direct Access methods provided by GPUDirect 2.0 have significantly reduced the effective communication requirements in parallel computer systems.

Graphics cards are connected to their host by the PCIe bus. GPUs will operate best when connected to a PCIe slot. Some motherboards have four PCIe slots and can host four cards. As there are dual-GPU graphics cards such as the GTX250, GTX350 or GTX400 a single host is capable of hosting up to eight GPUs. Figure 2 shows a single host connected to four single-GPU devices.

The GPU communication protocol is far more complex than that of the CPU. Different designs and operating systems schedules can make use of this in a way that is different from a CPU architecture but similar to the behavior expected of a GPU device.

Transfer through host

The transfer method is shown in Figure 5. An area of host memory is allocated and the data transfer occurs using a CPU function call.

Direct Transfer

The new mGPU-Direct 2.0 technology supported by Fermi architecture GPUs allows the unnecessary extra transfer through host memory to be eliminated. It allows data to be transferred directly from one GPU device to another through the PCIe bus. This direct transfer-free approach is shown in Figure 6. It is noted that although this data transfer does not involve the host memory it still initiated by a function call from the host.

Direct Access

The other option for inter-device communication that GPUs via the GPUDirect-RDMA protocol is called Direct Access. With this communication method the host is not involved in the transfer at all. Rather than a memory copy being initiated by the host to copy a segment of data from one device to the other, Direct Access allows the threads on one device to access data stored in the memory of another device. This memory access method with a thread on GPU reading a single memory location from GPU is shown in Figure 7.

mGPU Communication

The Direct Transfer and Direct Access methods provided by GPUDirect 2.0 have significantly reduced the effective communication requirements in parallel computer systems.

Graphics cards are connected to their host by the PCIe bus. GPUs will operate best when connected to a PCIe slot. Some motherboards have four PCIe slots and can host four cards. As there are dual-GPU graphics cards such as the GTX250, GTX350 or GTX400 a single host is capable of hosting up to eight GPUs. Figure 2 shows a single host connected to four single-GPU devices.

The GPU communication protocol is far more complex than that of the CPU. Different designs and operating systems schedules can make use of this in a way that is different from a CPU architecture but similar to the behavior expected of a GPU device.