Modeling and Performance Analysis in GPU Computing

Taylor Sasser

November 23, 2023

1 Modeling

- Allocate page-locked memory and retime PCI-express latency and bandwidth.
- Build a runtime model of polynomial expansion. (Express performance in coefficients calculated per second.)

1.1 Runtime Model of the PCI Bus

Given a PCI-Express bandwidth of 11 GB/s and a latency of approximately 200 ns, the runtime for data transfer across the PCI bus can be modeled as the sum of the fixed latency and the time taken to transfer the data. The formula for this model is as follows:

$$\text{Runtime} = 200\,\text{ns} + \frac{\text{DataSize}}{11\,\text{GB/s}}$$

In this model, *DataSize* represents the amount of data being transferred.

2 Pipelined Polynomial Expansion

- Transform the code to compute the polynomial expansion by chunks transferred asynchronously using multiple streams to overlap communication and computation.
- *Hint: Start by writing with a version that uses a single stream as it is easier. A good way to check if the code works is to verify it works on arrays larger than the size of the GPU.*
- Tune block size manually to get better performance. Compute for different degrees (0 to 100) and sizes of the array (100 to 10B) the performance obtained by using multiple streams.
- How does the performance achieved relate to the model?

2.1 Tuning Block Size and Performance Analysis

The optimal block size for the polynomial expansion computation was determined to be 1024. With this block size, the performance of the polynomial expansion was observed to be approximately 16 TFLOPS, or about 80% of the 19.2 TFlops the GPU is able to do. This was calculated with an input of 35000000 and a degree of 30000

3 Multiple GPUs

- Adapt the runtime model to account for multiple GPUs.
- Adapt the code to share the work across multiple GPUs.
- Compute for different degrees (0 to 100) and sizes of the array (100 to 10B) the performance obtained by using two GPUs.
- How does the performance achieved relate to the model?

3.1 Performance with Two GPUs

When the computation was distributed across two GPUs, there was a notable increase in performance for large degree polynomials, achieving about 27 TFLOPS. This improvement is consistent with the expectations set by our run time model, indicating that the use of multiple GPUs efficiently scales the computation.