PA 2 Solution Set

April 17, 2012

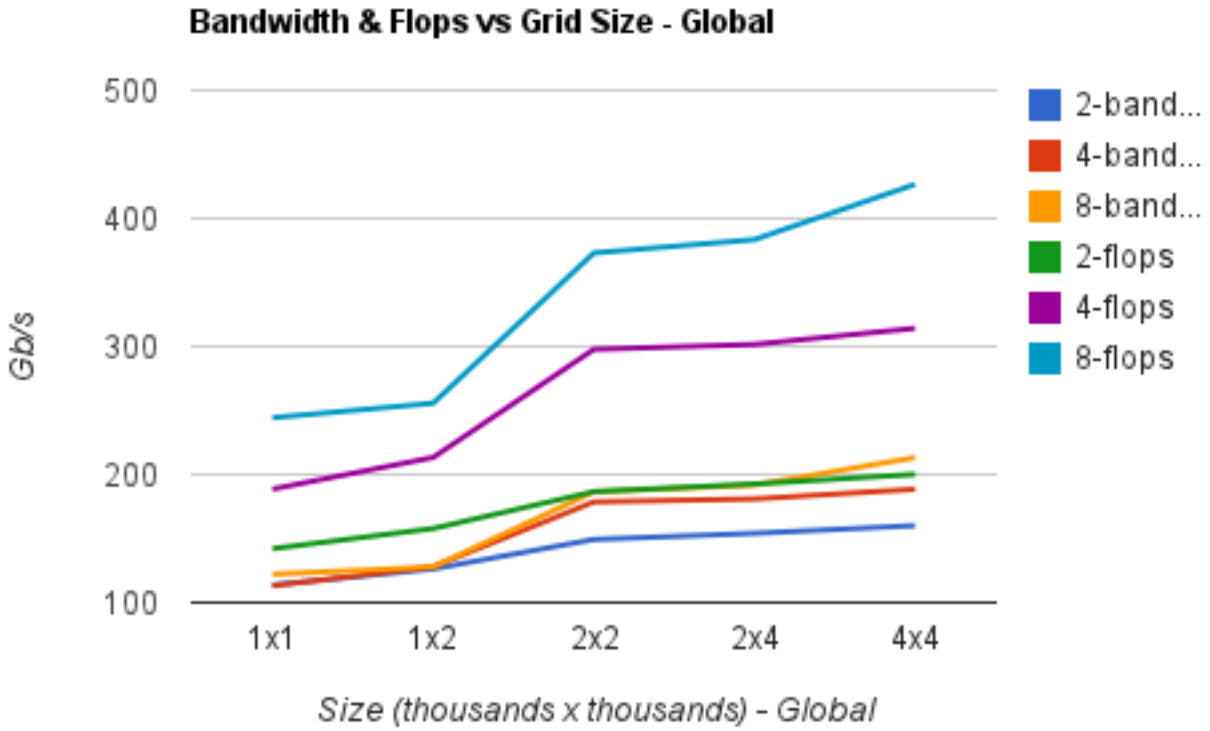
Correctness

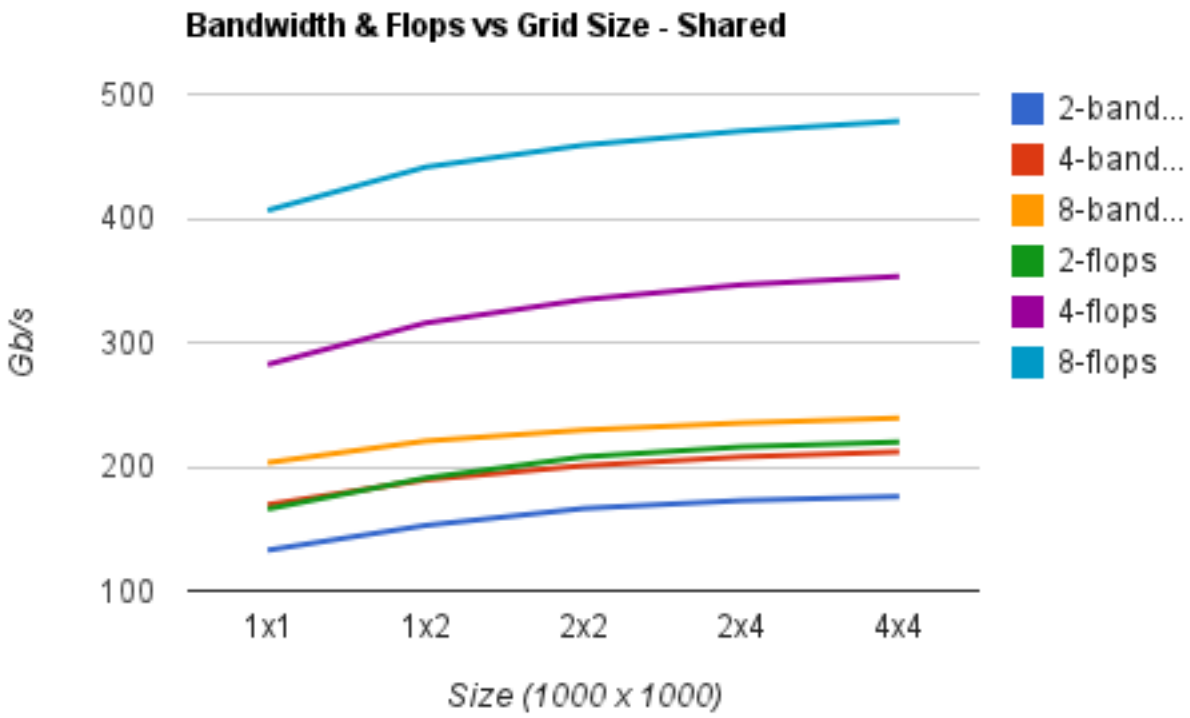
See CUDA solutions for a correct implementation

Analysis

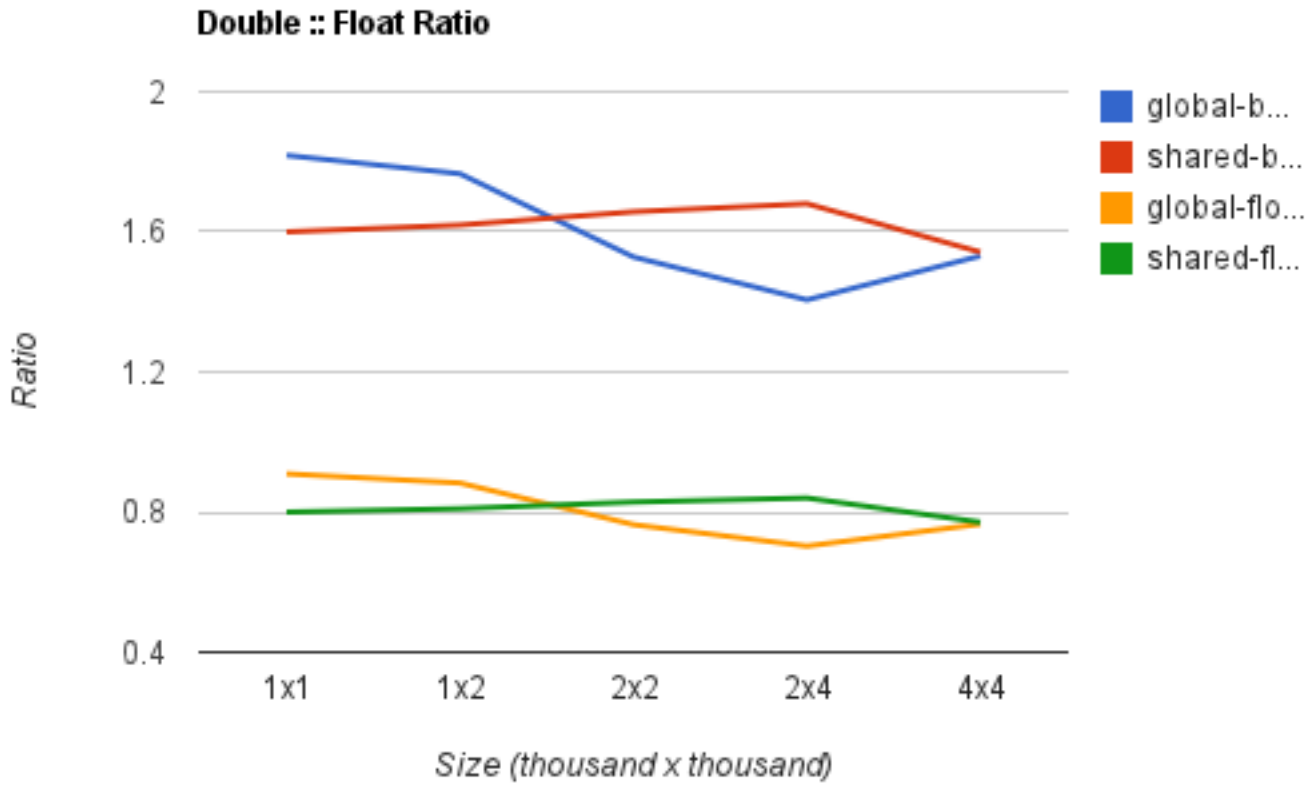
There are a few important observations :

1. We can see in the global memory case there is a sharper rise to the flops computation. This indicates that our memory bandwidth is the bottleneck, and
2. The order corresponds to higher performance in both categories. This is likely due to caching effects in the global memory case and shared memory by virtue of being shared memory.
3. Shared memory has similarly structured graphs, but higher flops and bandwidth. Shared memory is optimized around the idea of spatio-temporal locality, so this intuitively makes sense.
4. Double has better bandwidth but lower flops. Double-precision computation must compute more digits, so it is understandable that it has worse flops. Its memory bandwidth is higher because doubles have more bytes than floats, but really they are both serving the same number of requests.





Here we compare the ratio of performing these operations on float typed numbers (single precision) versus double typed (double precision).



Double has better bandwidth but lower flops. Double-precision computation must compute more digits, so it is understandable that it has worse flops. Its memory bandwidth is higher because doubles have more bytes than floats, but really they are both serving the same number of requests.