General-Purpose Computation on Graphics Hardware Seminar Lab Assignment

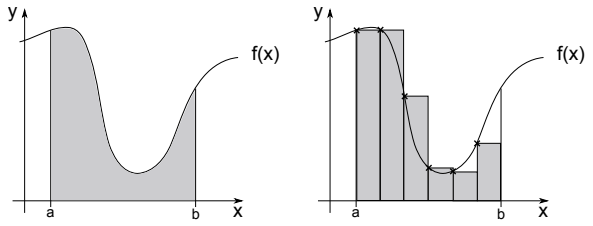
May 29th 2010

1 Numerical Integration

A simple algorithm for calculating the numerical value of a definite integral is the rectangle rule. It approximates the numerical value of an integral using a rectangle:

$$\int_a^b f(x)\,dx \approx (b-a)\,f\left(\frac{a+b}{2}\right)$$

By reducing the distance between a and b and summing the partial definite integrals a reasonable approximation of the definite integral can be calculated for many functions $f(x)$. The following image illustrates the idea:

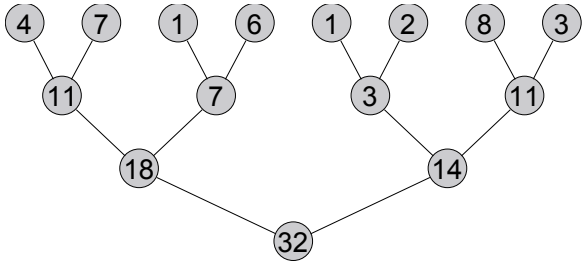


The *integration* CUDA project contains an incomplete implementation of the numerical integration algorithm.

1. Complete the CUDA implementation of the algorithm and compare the results to the CPU implementations.
2. Add functionality to benchmark the runtime of the algorithm.
   - remember that kernel calls are asynchronous
   - don't bench the first call to a kernel as it entails setup overhead
3. Integrate different functions and compare the kernel runtime.
4. Vary the number of blocks and the number of threads per block and compare the kernel runtime.

2 Reduction

In the current implementation the results of the numerical integration have to be added by the CPU to get the final result. This is the equivalent to a reduce function. Reduce is an important parallel primitive. The following image illustrated the principle of parallel reduce (summation):



The *reduction* CUDA project contains an inefficient implementation of reduce.

1. Try to optimize the reduce function. Two important performance factors are divergence and shared memory bank conflicts.

2. Compare the performance of the reduce function to the GPU peak performance. What metric is suitable for this comparison?