Christian DeLozier – CIS 565 Project Proposal

Moving the STL into the Heterogeneous Era

Prior to the rise of parallel and GPU computing, the C++ STL was designed for serial execution. With the advent of these new technologies, the performance of the STL could be accelerated in the presence of a GPU. Currently, each individual programmer must explore the performance trade-offs of serial CPU execution and GPU execution. Through parallel acceleration, the STL could continue to provide abstracted performance gains for C++ programmers by choosing the best execution for them. For my project, I plan to modify portions of the STL to make use of parallel computation when beneficial.

At the PLDI 2011 "Crazy Ideas" session, a Microsoft presenter noted that his team had been able to significantly accelerate the STL library functions using CPU vector operations. In a similar manner, GPU acceleration might be used to boost the performance of the STL. As a starting point, the Thrust C++ library provides matching GPU algorithms for many of the STL algorithms, and the LLVM team provides a clean implementation of the STL libraries. I will refactor these libraries to accelerate performance using the Thrust libraries when conditions, such as the size of the input data, will allow for a performance increase. To identify the conditions for increased performance, I will analyze the performance of the STL and Thrust libraries under various STL microbenchmarks that I will develop.

Thrust does not implement common STL containers such as list and map, but STL algorithms over these data structures might also be accelerated. As an example, a list could be mapped to a vector, execute on the GPU using the Thrust libraries, and then be mapped back to the original list. It may also be possible to further accelerate computations in a similar manner to Array Building Blocks, which queues computations and lazily executes them when data is first requested. In this way, we could avoid copying data to and from the GPU more frequently than necessary.

To evaluate the success of my project, I would measure the performance of the modified STL under various microbenchmarks and the C++ benchmark suite offered by Adobe.