

Jacob Chang
Alex Solomatnikov
Jae-Wook Lee
Nate Hill

Mapping Stream Stencils to Imagine

We are planning to investigate compilation tools from Brook to Stream C for our project, focusing on the intercluster communication required by stencils. The way we plan to approach the problem is to have a set of general communication mechanisms as well as templates that are optimized for certain stream access patterns. When available we will select one of the more optimal communication templates.

The input stream access stencil can be read from the Brook code through the compiler. However, in order to keep our project simple, we are going to manually input the stencil instead of relying on the compiler to extract the pattern for us.

The first part of the project would be to translate simple stencils to KernelC. A good place to start will be the 6-tap and 24-tap fir Brook code. There might be other StreamC programs that have been written we might want to look at to see what kind of communication it uses. A good performance matrix here would be to see how many cycles it takes to perform the communication required for each template. Then we can rank the different template based on its communication performance. Also, as we find optimizations, we need to bin what stencils each template can support.

The second part of the project would be to come up and implement a script that automatically select one of these communication template depending on the input stream access pattern. The output would be a skeleton of the code including the communication making the stencil available.