Proposal:

Predetermined Source-Source Translations via a new type of Compiler

Goal:

Develop a source-to-source compiler through the use of stateless predefined templates.

Problem Statement:

Heuristic compiler optimizations work well enough but fail to take advantage of the emerging complex acceleration instruction sets such as Intel AVX2 and AMD XOP. The primary failure point is that heuristics work via a set of generalized transformations on an intermediate representation, usually on an abstract syntax tree, and these transformations are very complicated to alter without affecting backwards compatibility. With the aid of a template matching algorithm and many handwritten templates, I propose a trivial method of compilation that also allows greater flexibility with optimizations for emerging complex instruction sets.

Implications:

In addition to the heuristic optimizations usually applied, it may be possible to increase average computation resource load by several times via a single pre-pass. By applying concepts from this project, compiler codebases could shrink by a large percentage while offering vastly improved optimizations. In addition, if the instruction-level efficiency is similar to the heuristic implementation then the idea of a template compiler could provide a very simplified interface for a language to support the latest acceleration instruction sets.

Milestones:

1. Decide on template-match algorithm.
2. Write simple template-matching database implementation.
3. Create several templates from primary source to secondary source.
4. Decide on the template compiler implementation language.
5. Create template compiler.
   1. Create parser/tokenizer.
   2. Integrate with template-match database.
   3. Invoke templates to translate primary source to secondary source.
   4. Verify that generated source is compliable by 3rd party compiler.
   5. Verify that generated source is executable after 3rd party compilation.
6. Overhaul compiler to allow for optimizing the output source for use with current and future acceleration instruction sets.
7. Write project report.