# Software Based Speculative Multi-Threading System

## Goals

The goal is to provide an easy to use toolchain that automatically parallelizes sequential programs at the cost of additional runtime overhead. A speculative multi-threaded (SMT) system can leverage its runtime checks to extract parallelism out of programs which are hard or impossible to parallelize by conventional automatic parallelization systems. It also requires very little user intervention.

## Program Breakdown

The system is broken down in 4 modules which will be developed iteratively:

1. The runtime speculative system.
   1. Spawns threads based on the available resources.
   2. Dynamically assigns tasks to the spawned threads.
   3. Performs checks to detect dependency violations.
   4. Rollback any modifications done by the threads involved in the violations.
2. Code generation system.
   1. Analyses the code looking for good parallelization candidates.
   2. Injects library calls into the target source code.
   3. Implemented using the clang front-end for LLVM.
3. LLVM optimization passes
   1. Applies optimizations that are specific to an SMT system.
4. Feedback system.
   1. Analogous to profile-guided optimizations.
   2. At runtime the library gathers information that can then be fed back into the compiler process of module 2 and 3 to make better decisions about the code.

## PFE Scope

It's pretty obvious that the whole system is too much work to implement by the end of the PFE. The idea is then to finish the implementation of the first module (which happens to contain the riskiest pieces) and hopefully get started the second module using only simple programming constructs. The module 3 and 4 are mostly going to be used as fodder for documents (Vision, SRS, Architecture, etc.) if needed.

## Useful links

http://en.wikipedia.org/wiki/Speculative\_execution *(Good example of a speculative system)*

http://en.wikipedia.org/wiki/Speculative\_multithreading *(included for the document references)*

http://openmp.org/wp/ *(Architecture of the system is inspired by OpenMP)*

http://llvm.org/

http://clang.llvm.org/

http://en.wikipedia.org/wiki/Profile-guided\_optimization