Orthogonal Scheduling of Stencil Computations with Chapel Iterators

Student: Ian Bertolacci
Advisors: Michelle Mills Strout Catherine Olschanowsky

Problem
Implementing stencil computations is already a difficult task. The addition of specifying iteration schedules with both good parallelism and high data locality introduces a new dimension of difficulty, and increases the load on the developer.

Compilers already exist to automate the task of optimization, but may lack the information required to generate the most optimal code. Our research investigates whether a library can be developed that gives the programmer a more optimal schedule in a plug-and-play manner without introducing complexity into their code.

Programmability
We want to transform our original schedule:

```chapel
for t in timeRange do
  forall (x, y) in spaceDomain do
    computation(t, x, y);
```

into a faster schedule:

```chapel
forall (t, x, y) in slicedDiamondIterator() do
  computation(t, x, y);
```

while avoiding code with equivalent performance, but agony to develop, maintain, or understand:

Future Work
- Fix performance gap between Chapel and OpenMP
- Creation of Chapel tiling iterators library
- Optimum tile size discovery algorithm
- Generic, N-dimensional, dependency realizing iterators
- OpenMP C parallel iterators

References