0.0.1 Auto-parallelizing Pure Functional Language System

Report by:	Kei Davis
Participants:	Dean Prichard, David Ringo, Loren Anderson, Jacob Marks
Status:	active

The main project goal is the demonstration of a light-weight, higher-order, polymorphic, pure functional language implementation in which we can experiment with automatic parallelization strategies, varying degrees of default function and constructor strictness. A secondary goal is to experiment with mechanisms for transparent fault tolerance. We do not consider speculative or eager evaluation, or semantic strictness inferred by program analysis, so potential parallelism is dictated by the specified degree of default strictness and any strictness annotations.

Our approach is similar to that of the Intel Labs Haskell Research Compiler: to use GHC as a front-end to generate STG (or Core), then exit to our own back-end compiler. As in their case we do not attempt to use the GHC runtime. Our implementation is *light-weight* in that we are not attempting to support or recreate the vast functionality of GHC and its runtime. This approach is also similar to Don Stewart's except that we generate C instead of Java.

Current Status

Currently we have a fully functioning serial implementation and a primitive proof-of-design parallel implementation.

Immediate Plans

We are currently developing a more realistic parallel runtime. Bridging the gap between GHC STG (or Core) to our STG representation will be undertaken starting June 2016. An instrumentation framework will be developed in summer 2016.

Further reading

A project web site is under construction.

Undergraduate/post-graduate Internships

If you are a United States citizen or permanent resident alien studying computer science or mathematics at the undergraduate level, or are a recent graduate, with strong interests in Haskell programming, compiler/runtime development, and pursuing a spring, fall, or summer internship at Los Alamos National Laboratory, this could be for you.

We don't expect applicants to necessarily already be highly accomplished Haskell programmers—such an internship is expected to be a combination of further developing your programming/Haskell skills and putting them to good use. If you're already a strong C hacker we could use that too.

The application process requires a bit of work so don't leave enquiries until the last day/month.

Term	Application Deadline
Summer 2016	Closed
Fall 2016	May 31, 2016
Spring 2017	Approx. July 2016
Summer 2017	Approx. January 2017
Fall 2017	Approx. May 2017

Email me at kei (at) lanl (dot) gov if interested in more information, and feel free to pass this along.