0.0.1 Auto-parallelizing Pure Functional Language System

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

The main project goal is the demonstration of a lightweight, higher-order, polymorphic, pure functional language implementation in which we can experiment with automatic parallelization strategies and 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, using GHC as a front-end to generate STG, 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. The most recent major development was the “bridge” between GHC and our system. Thus we can now compile and run Haskell programs with simple primitive and algebraic data types.

Immediate Plans

We are currently developing a more realistic parallel runtime. Tentatively the fault tolerance mechanism is scheduled as a Master's thesis project starting summer 2017.

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. Dates for terms beyond summer 2017 are best guesses based on prior years.

Term	Application Opening	Deadline
Summer 2017	Open	Jan 2017
Fall 2017	Jan 2017	May 2017
Spring 2018	May 2017	Jul 2017

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

Further reading

Email me as above for the Trends in Functional Programming 2016 paper about this project.