Compiler, Heterogeneous Parallel Computing and Mathematical Optimization

Lanting Guo

September 5, 2015

Background

- Foucus on machine learning Data Exploration ==> Model ==> Loss Function / || || Iterative || || / Accuracy <== code <== Optimization Algorithm
- three mountatians ==> one mountatian
- MLer() = compute_ability(HPC) + algorithm(OPT) + model(DL, PGM, ML, Boosting, Ensemble etc)
- iterative, coordinate ascent and active learning:
  - iterative: again, again, again, ...
  - coordinate ascent: a simple heuristic algorithm, optimize one while fixed the others.
  - active learning: learn what you like, learn what you are capable of, and valuable of...
- minimize: the height of three mountatians subject to:
  sum(time_i) = 10000hours
  keep other states relatively stable and persistant

Compiler

- Why Compiler?
  - deeper understanding about programming languages
  - part of it related to convex optimization and DL softwares: theano, MShadow
  - coding better
- LLVM
- kaleidoscope
- Julia
- Cxx.jl

LLVM

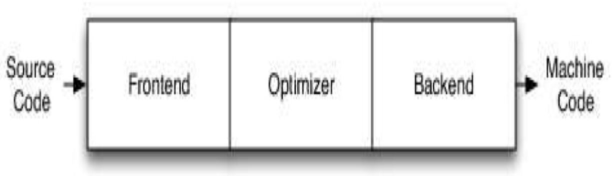
LLVM, `http://llvm.org`

- What: formerly "Low Level Virtual Machine," today general purpose compiler infrastructure
- Who: many contributors from Apple, Google, Intel, Mozilla, Julia, etc. Used by Clang, Rust, Swift, Emscripten, WebKit (Safari)
- When: originally Chris Lattner's Masters and Ph.D theses, circa 2003
- License: University of Illinois / NCSA (permissive, BSD-style)
- Written in: `C++`
- Use in Julia: just in time compiler

Classical compiler design -1

From "The Architecture of Open Source Applications,"
`http://www.aosabook.org/en/llvm.html`

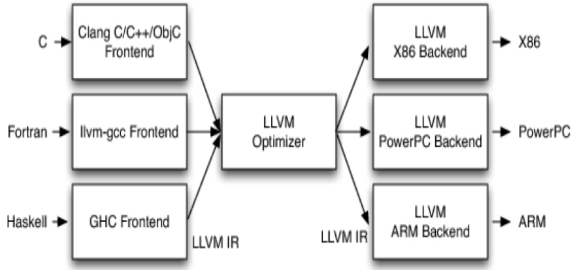
- Basic 3 phase compiler



- One input language, one target architecture

Classical compiler design -2

- Modular compiler design



- Reuse core components across multiple languages and architectures
- LLVM intermediate representation (IR)
  - Sort of like "cross-platform assembly"
  - Try out `@code_llvm` in Julia

introduction to Julia internal

Heterogeneous Parallel Computing

- julia: multi-thread, multi-task MC demo
- CUDA C: Monto Caro demo
- ClusterManagement.jl

Mathematical Optimization

- SVM
- L1 regulation

an machine learning example

- First kaggle TOP10%