

Compiler, Heterogeneous Parallel Computing and Mathematical Optimization

Lanting Guo

September 7, 2015

- Focus on machine learning

```
Data Exploration ==> Model ==> Loss Function
/\                               ||
||      Iterative              ||
||                               \/\
Accuracy <== code <== Optimization Algorithm
```

- $\text{MLer}() = \text{compute_ability}(\text{HPC}) + \text{algorithm}(\text{OPT}) + \text{model}(\text{DL, PGM, ML, Boosting, Ensemble etc})$
- 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` in 3 years
keep other states relatively stable and persistent

- Why Compiler?
 - the models, theory and algorithms can be applied to a wide range of problems in software design and development
 - deeper understanding about programming languages
 - part of it related to convex optimization and DL softwares: theano, MShadow
 - coding better
- LLVM
- kaleidoscope
- Julia Internal
- Cxx.jl

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

From "The Architecture of Open Source Applications,"

<http://www.aosabook.org/en/llvm.html>

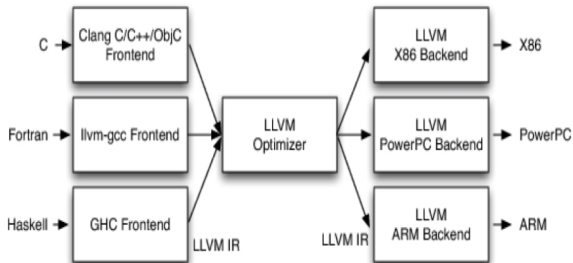
- Basic 3 phase compiler



- 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

kaleidoscope at my Github

Heterogeneous Parallel Computing

- julia: multi-thread, multi-task
- CUDA
- ClusterManagement.jl

Mathematical Optimization

- SVM
- L1 regulation

an machine learning example

- First kaggle TOP10%

END

Thank you!