Multi-ML cross-platform meta-programming at scale

using compiler tech and C

Samuel Marks, PhD
Computer Science Engineering

Annual review 2023-08-14

Introduction

Programming solutions
Nothing I build will be popular. Neither a new language nor a new framework. Thus, rather than attempt to solve problems of dev speed, scale and quality this way; these are solved through automated interoperability.

Deployment (multicloud, DevOps)
Deploying software is complicated by the multitude of cloud vendors, operating systems, and depth of dependencies; including databases and libraries.

Fractionated ML
The ML field is moving too rapidly. There is little confidence to be found in SOTA claims (e.g., in medicine).

Biography

Samuel Marks

Academic

Holds PhD from the University of Sydney.

Commercial

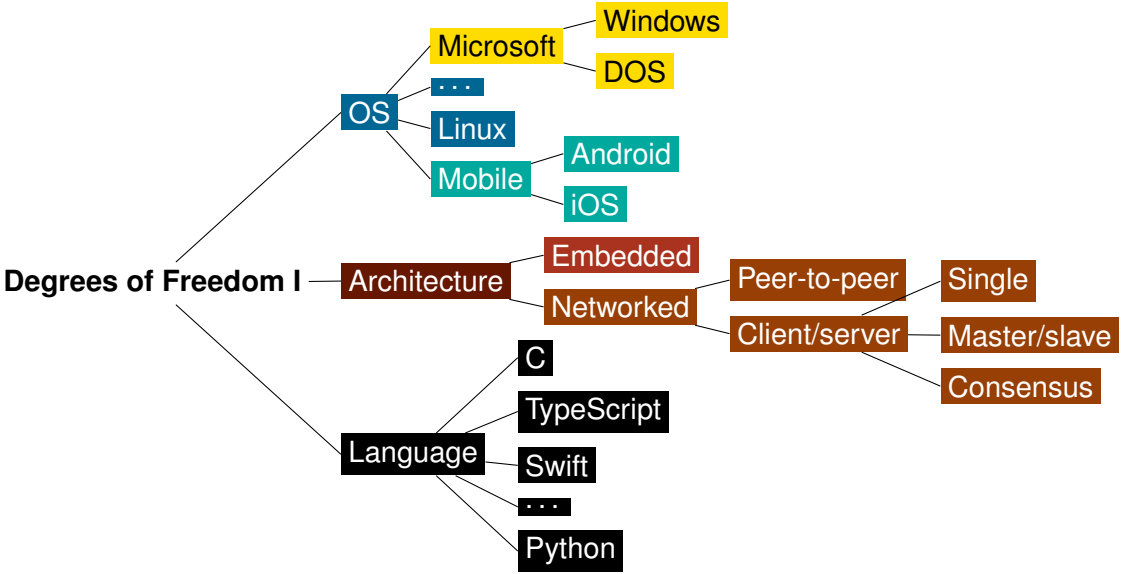
Delivered many dozens of projects for dozens of companies over 10+ years.

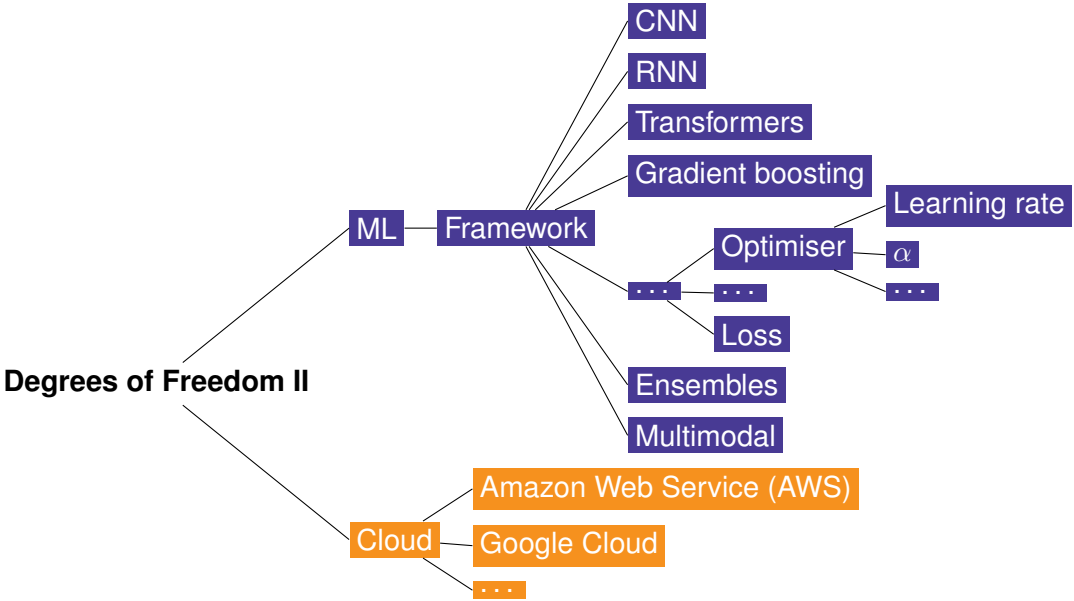
Charitable

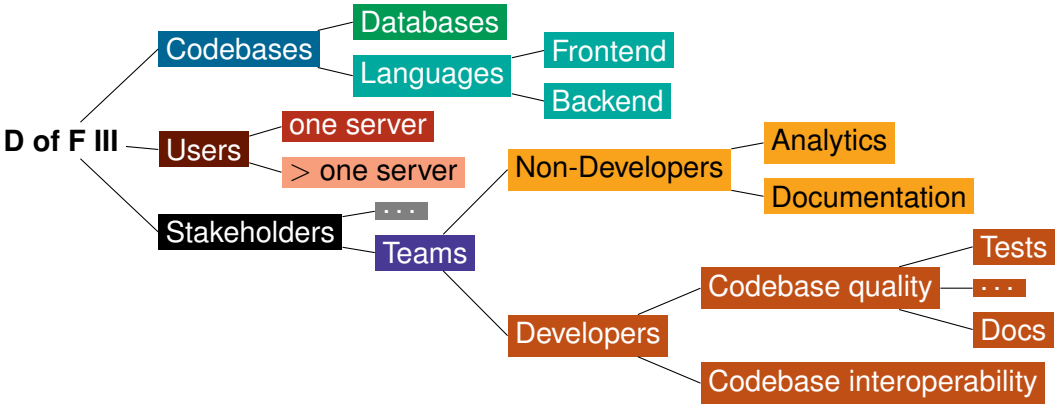
Working on facilitating mass screening programmes for blinding eye diseases.

History and definitions

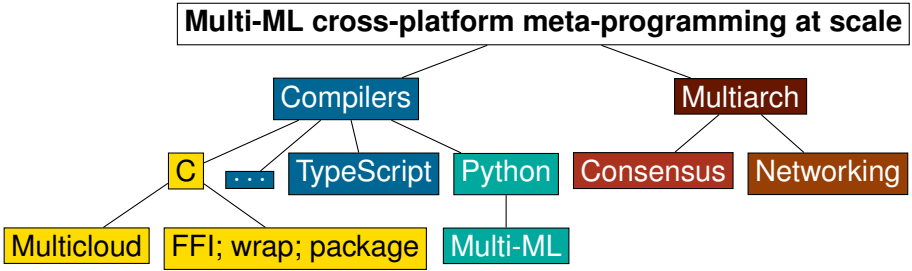
Open thesis PDF


Degrees of Freedom I
OS
Microsoft
Windows
DOS
...
Linux
Mobile
Android
iOS
Architecture
Embedded
Networked
Peer-to-peer
Single
Client/server
Master/slave
Consensus
Language
C
TypeScript
Swift
...
Python


Degrees of Freedom II
ML
Framework
CNN
RNN
Transformers
Gradient boosting
Optimiser
Learning rate
α
...
...
...
Loss
Ensembles
Multimodal
Cloud
Amazon Web Service (AWS)
Google Cloud
...


D of F III
Codebases
Databases
Languages
Frontend
Backend
Users
one server
> one server
Stakeholders
. . .
Teams
Non-Developers
Analytics
Documentation
Developers
Codebase quality
Tests
. . .
Docs
Codebase interoperability

Thesis ontology



Interoperability

> *The ability of two or more systems or components to exchange information and to use the information that has been exchanged.*
> — IEEE Standard Computer Dictionary (1990)

Aside from using same language [runtime], networking, and executable chains; there are two basic ways of achieving interoperability:

DSL (types)

- IDL (1983)
- ASN.1 (1984)
- SGML (1986)
- XML Schema (2001)
- JSON Schema (2009)
- ...

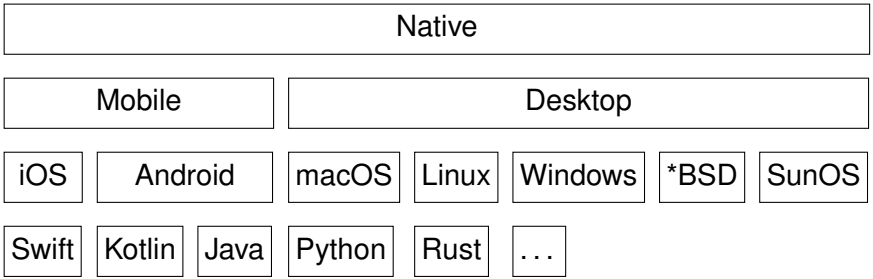
DSL (API)

- SOAP (1998)
- WSDL (2000)
- WADL (2009)
- OpenAPI (2011)
- AsyncAPI (2016)
- ...

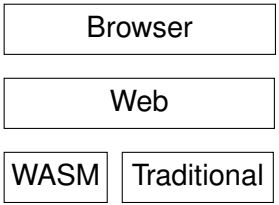
C

- `extern` to embed within C++
- Binary level
- Library level
- ABI
- FFI for most every other language

Languages and targets



Languages and targets



← all 7 can target any OS (except Swift on Android, and Java on iOS)

C

A popular programming language invented in 1972. With support across:

Operating Systems

- Android
- iOS
- Linux
- Windows
- DOS
- web
- macOS
- SunOS
- *BSD
- . . .

Programming Languages

Foreign Function Interfaces

- Python
- Rust
- Swift
- PHP
- Crystal
- Java
- Erlang
- Haskell
- Nim
- . . .

Multi-ML

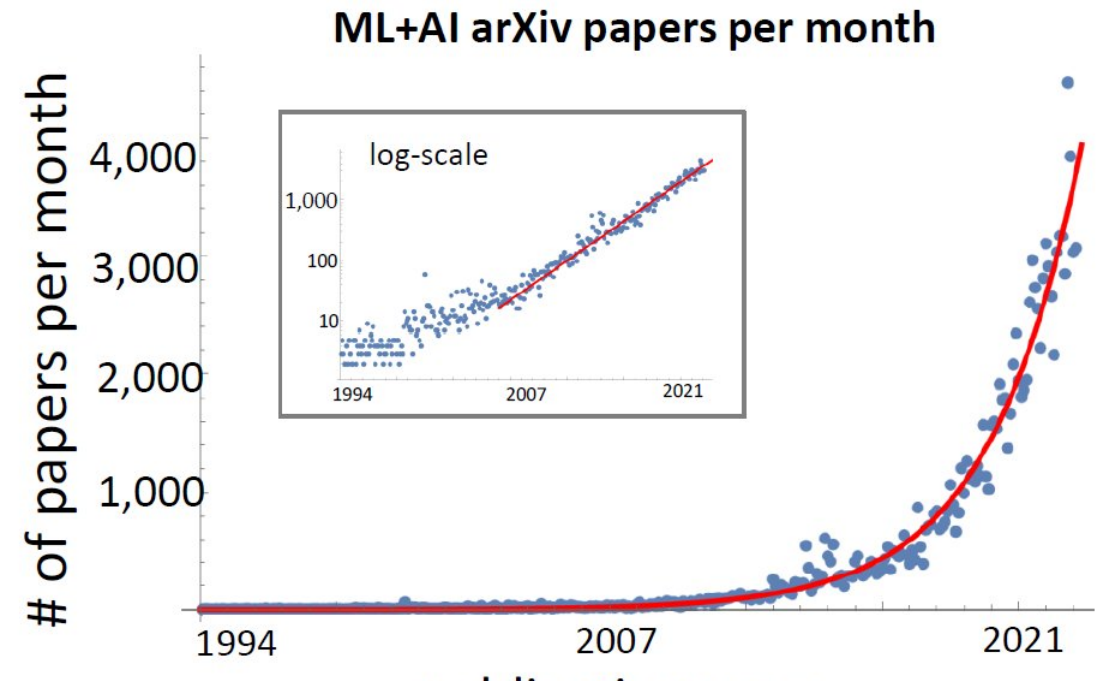
NOTE: The landscape is changing. I am a top 10 contributor to Google's [TensorFlow] Keras. A month ago they released a cross-compatible version that makes interchangeable PyTorch, JAX, and TensorFlow.

Multi-ML

There are 10 commonly used ML frameworks. Each have different ecosystems, and when a new research paper—or industry project—is released, they (usually) target just one framework.

My new multi-ML framework is created by applying my `cdd-python` compiler to 10 different popular ML frameworks (at the source-code level). This exposes CLIs, REST APIs, database tables and other useful layers.

Now, given a problem (e.g., determine best dataset for my new optimiser, or determine best [AUCROC] model for my new dataset), the framework will optimise across a *search space* traversing permutations of parameters (e.g., optimiser, loss function) and hyperparameters (e.g, α, β, learning rate). Where *search space* can include everything that the ML ecosystem has to offer.


ML+AI arXiv papers per month
# of papers per month
4,000
3,000
2,000
1,000
1994
2007
2021
log-scale
1,000
100
10
1994
2007
2021

Multi-ML - Future work $\frac{I}{II}$ (post-PhD; or by others)

Take arbitrary repos with Python packages or simple Notebooks & automatically:

0. Find and clone candidate repositories (e.g., from the arXiv);
1. Make OS independent;
2. Remove absolute paths (e.g., to weight files);
3. Format and autolint;
4. Add type hints;
5. Separate steps to be compatible with ensemble use-cases, e.g., move the model construction to its own function, and constants [like kernel sizes] to a consistent section;
6. Send pull-request / merge-request to repository;
7. If PR is accepted, add new model, optimizer, loss function, or other relevant 'thing' to this multi-ML meta-framework's search-space;
8. Publish (online) benchmarks of this new 'thing' against similar 'thing's on a variety of different datasets.

Multi-ML - Future work // // (post-PhD; or by others)

0. Automatically systematically review articles [with public datasets] coming through different research fields;
1. run their claims against the entire search-space of this multi-ML meta-framework;
2. (if improvement detected) write up a research paper with a new claim.
3. (if improvement detected) Open-source the (new) result with clear replication steps.

Add support for new ML frameworks.

Analytics and dashboarding.

Multiclustering

Depending on ones use case, different architectures make sense, commonly they are:

- Embedded
- Client/server
  - Single server
  - Master/slave
  - Consensus
    - One master multiple slaves
    - Read-only slaves with multiple masters
    - Peer-to-peer

Focussing on multiclustering means not ignoring the scalability problems of software-engineering. Implementation TBD, and to be authored in C.

Multicloud and DevOps

There are a large number of cloud vendors with an exponentially larger number of DevOps tooling. Both complicate software development, maintenance, compliance, security, and debugging.

Multicloud
Created new cloud vendor client libraries in C. With focus on FFI interoperability and cross-platform compatibility, these can be called on any system and by any language or extensible framework.

DevOps
Portable cross-platform single application version managers are developed. Each can run in any configuration management tool (25+) and on any OS; and thus any cloud.

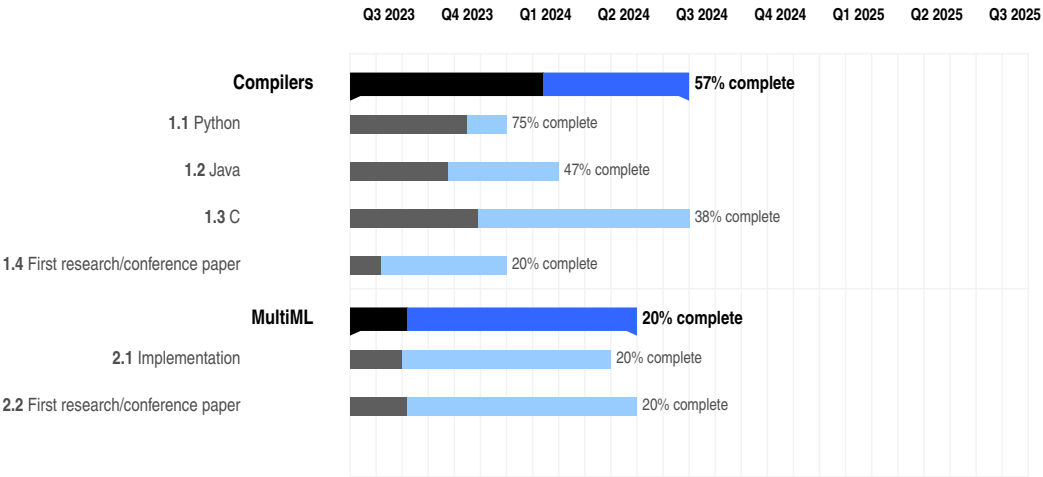
Future work

We can also add two columns in the slides.

Instrument compilers to trans bodies

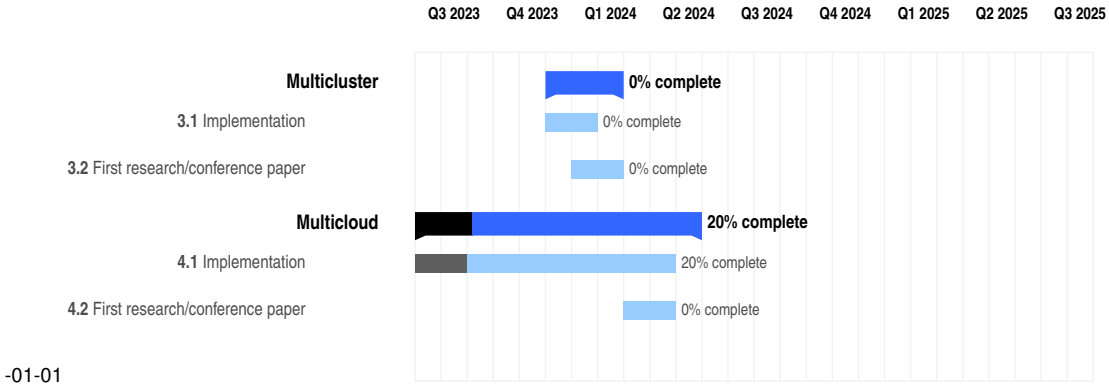
Auto read/write papers, replicate their result then improve [Multi-ML] then publish (all automatic)

Plan I/III



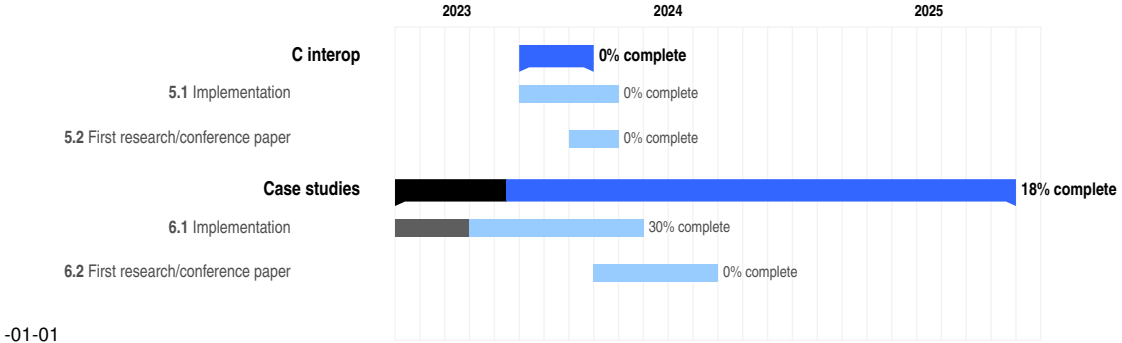
-01-01

Plan


Q3 2023
Q4 2023
Q1 2024
Q2 2024
Q3 2024
Q4 2024
Q1 2025
Q2 2025
Q3 2025
Multicluster
0% complete
3.1 Implementation
0% complete
3.2 First research/conference paper
0% complete
Multicloud
20% complete
4.1 Implementation
20% complete
4.2 First research/conference paper
0% complete

-01-01

Plan


2023
2024
2025
C interop
0% complete
5.1 Implementation
0% complete
5.2 First research/conference paper
0% complete
Case studies
18% complete
6.1 Implementation
30% complete
6.2 First research/conference paper
0% complete
-01-01

Acknowledgements

Dr Alan Blair for supervising. Wife and baby for muse and amusement.