Technical Advisory Council Meeting

February 27, 2020

THE LINUX FOUNDATION

LF AI

Antitrust Policy Notice

› Linux Foundation meetings involve participation by industry competitors, and it is the intention of the Linux Foundation to conduct all of its activities in accordance with applicable antitrust and competition laws. It is therefore extremely important that attendees adhere to meeting agendas, and be aware of, and not participate in, any activities that are prohibited under applicable US state, federal or foreign antitrust and competition laws

› Examples of types of actions that are prohibited at Linux Foundation meetings and in connection with Linux Foundation activities are described in the Linux Foundation Antitrust Policy available at http://www.linuxfoundation.org/antitrust-policy. If you have questions about these matters, please contact your company counsel, or if you are a member of the Linux Foundation, feel free to contact Andrew Updegrove of the firm of Gesmer Undergone LLP, which provides legal counsel to the Linux Foundation

Recording of Calls

This is a reminder that TAC calls are recorded and available for viewing on the TAC Wiki

THE LINUX FOUNDATION

LF AI

Reminder: LF AI Useful Links

Web site: lfai.foundation
Wiki: wiki.lfai.foundation
GitHub: github.com/lfai
Landscape: landscape.lfai.foundation or l.lfai.foundation
Mail Lists: https://lists.lfai.foundation

LF AI Logos: https://github.com/lfai/artwork/tree/master/lfai
LF AI Presentation Template:
https://drive.google.com/file/d/1eiDNJvXCqSZHT4Zk_-czASIz2GTBRZk2/view?usp=sharing

Events Page on LF AI Website: https://lfai.foundation/events/
Events Calendar on LF AI Wiki (subscribe available):
https://wiki.lfai.foundation/pages/viewpage.action?pageId=12091544
Event Wiki Pages: https://wiki.lfai.foundation/display/DL/LF+AI+Foundation+Events

Agenda

- Roll Call
- Approval of Minutes
- TAC Chairperson Election
- Invited Talk: CodeReef
- LF AI Updates
- Upcoming TAC Meetings
- Open Discussion

TAC Voting Members

Member	TAC Contact	TAC Email
Amdocs	Ofer Hermoni*	oferher@gmail.com
AT&T	Jack Murray	jfm@research.att.com
Baidu	Daxiang Dong	dongdaxiang@baidu.com
Ericsson	Nimisha Radia	nimish.radia@ericsson.com
Huawei	Huang Zhipeng	huangzhipeng@huawei.com
Nokia	Pantelis Monogioudis	pantelis.monogioudis@nokia.com
Tech Mahindra	Nikunj Nirmal	nn006444@techmahindra.com
Tencent	Juniping Du	junipingdu@tencent.com
Zilliz	Jun Gu	jun.gu@zilliz.com
ZTE	Wei Meng	meng.wei2@zte.com.cn
Acumos AI Project	Anwar Aftab	anwar@research.att.com
Angel Project	Fitz Wang	fitzwang@tencent.com
ONNX Project	Prasanth Pulavarthi	prasanth.pulavarthi@microsoft.com

* TAC Chairperson

Approval of Minutes

› Draft minutes from the February 13th, meeting of the TAC were previously distributed to the TAC members

› **Proposed Resolution:**

  › That the minutes of the February 13th meeting of the Technical Advisory Council of the LF AI Foundation are hereby approved

TAC Chairperson Election

› The annual TAC Chairperson election will begin in April
› Please begin to think about any interest in this leadership role
› Nominations will kick off mid April, followed by voting, and results by end of April

› General TAC and Chairperson details can be viewed within the LF AI Charter under section 7 here - A summary of the role:
  › Represent the TAC as a voting member on the Governing Board (will attend monthly meeting)
  › Lead TAC agenda and meetings with coordination among the TAC representatives and broader community
  › Attend sync meetings with LF AI staff to discuss overall TAC activities and planning
  › General representation of the TAC and the LF AI technical community

THE LINUX FOUNDATION

LF AI

Invited Talk: CodeReef

Grigori Fursin
grigori.fursin@codereef.ai

cKnowledge.org: accelerating the adoption of AI/ML in production with portable CK workflows and reproducible crowd-benchmarking

or our community attempt to bring order to AI/ML chaos

Grigori Fursin, CK author and CTO of CodeReef.ai

cKnowledge.org/partners

CK complements and interconnects popular tools and services rather than substituting them

github.com/ctuning/ck

CodeReef.ai/static/docs

Many groups are working to co-design efficient AI / ML / SW / HW stacks

AI hardware
• All major vendors (Google, NVIDIA, ARM, Intel, IBM, Qualcomm, Apple, AMD …)

AI models
Many groups in academia & industry (Google, OpenAI, Microsoft, Facebook …)

AI software
• AI frameworks (TensorFlow, MXNet, PyTorch, CNTK, Theano)
• AI libraries (cuDNN, libDNN, ArmCL, OpenBLAS)

AI integration/services
• Cloud services (AWS, Google, Azure ...)

Efficient AI/ML system must be very carefully co-designed

Hardware

Algorithms

Models

Data sets

Libraries
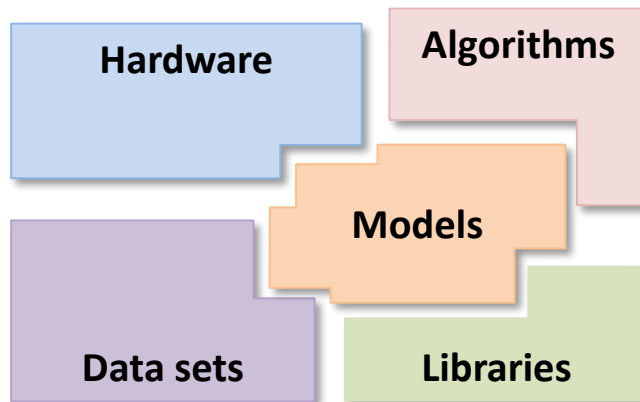
for various form factors
(IoT, mobile, data centers)

while trading off multiple constraints
(accuracy, speed, energy, size, costs)

and maximizing ROI
(faster time to market, R&D sustainability, much better than all competitors)

Helping the society

Healthcare
Agriculture
Finances
Automotive
Aerospace
Meteorology
Retail
Robotics
…

AI hardware
- All major vendors (Google, NVIDIA, ARM, Intel, IBM, Qualcomm, Apple, AMD …)

AI models
Many groups in academia & industry (Google, OpenAI, Microsoft, Facebook …)

AI software
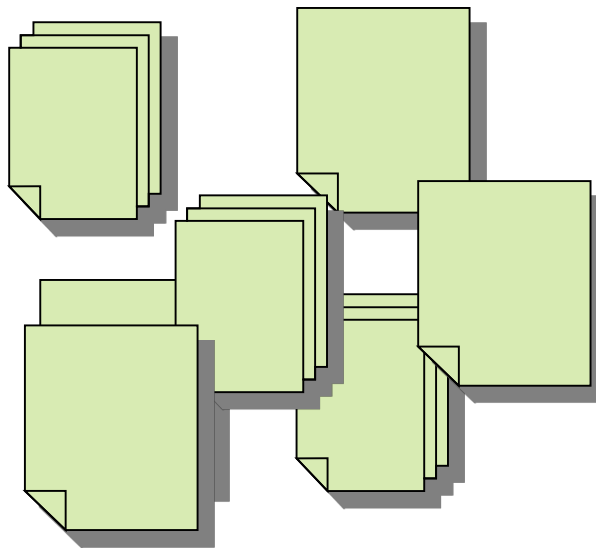- AI frameworks (TensorFlow, MXNet, PyTorch, CNTK, Theano)
- AI libraries (cuDNN, libDNN, ArmCL, OpenBLAS)

AI integration/services
- Cloud services (AWS, Google, Azure …)

Numerous papers with ad-hoc code

Numerous models, data sets, benchmarks, libraries and tools

Multiple competitions focusing mostly on accuracy (Kaggle, DawnBench)

A few benchmarks an competitions focusing on optimizing other metrics besides accuracy:

LPIRC, MLPerf

Helping the society

Healthcare
Agriculture
Finances
Automotive
Aerospace
Meteorology
Retail
Robotics
…

90K+ AI / ML / SW / HW papers are published each year!

AI hardware
• All major vendors (Google, NVIDIA, ARM, Intel, IBM, Qualcomm, Apple, AMD ...)

AI models
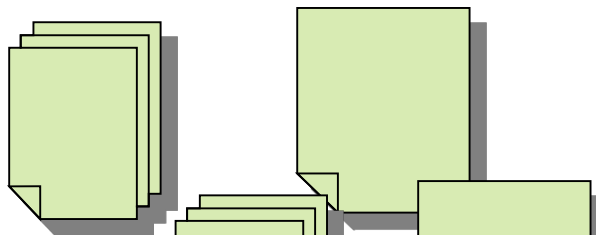Many groups in academia & industry (Google, OpenAI, Microsoft, Facebook ...)

AI software
• AI frameworks (TensorFlow, MXNet, PyTorch, CNTK, Theano)
• AI libraries (cuDNN, libDNN, ArmCL, OpenBLAS)

AI integration/services
• Cloud services (AWS, Google, Azure ...)

Numerous papers with ad-hoc code

Can we now co-design efficient SW/HW/ML stacks and use them in production to support real-world applications?

Need common, scalable, interfaces, benchmarks, libraries and tools

Multiple competitions focusing mostly on accuracy (Kaggle, DawnBench)

A few benchmarks an competitions focusing on optimizing other metrics besides accuracy:

LPIRC, MLPerf

Helping the society

Healthcare
Agriculture
Finances
Automotive
Aerospace
Meteorology
Retail
Robotics
...

Adoption of novel AI / ML techniques in production is extremely slow

AI hardware
• All major vendors
(Google, NVIDIA, ARM,
Intel, IBM, Qualcomm,
Apple, AMD …)

AI models
Many groups in
academia & industry
(Google, OpenAI,
Microsoft, Facebook …)

AI software
• AI frameworks
(TensorFlow, MXNet,
PyTorch, CNTK, Theano)
• AI libraries
(cuDNN, libDNN, ArmCL,
OpenBLAS)

AI integration/services
• Cloud services
(AWS, Google, Azure …)

• Technological chaos: continuously
changing algorithm/model/SW/HW stack

• Non-representative / outdated training sets

• No common experimental frameworks
and established methodologies
which can adapt to this chaos

• Numerous reproducibility issues

• Very little artifact reuse in 1000+ ML papers

• Very little tech. transfer from academia
(toy examples and too many papers)

Helping the society

Healthcare
Agriculture
Finances
Automotive
Aerospace
Meteorology
Retail
Robotics
…

Adoption of novel AI / ML techniques in production is extremely slow

AI hardware
• All major vendors (Google, NVIDIA, ARM, Intel, IBM, Qualcomm, Apple, AMD …)

AI models
Many groups in academia & industry (Google, OpenAI, Microsoft, Facebook …)
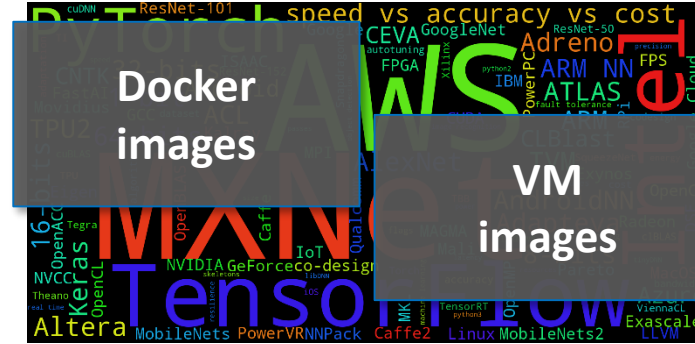
AI software
• AI frameworks (TensorFlow, MXNet, PyTorch, CNTK, Theano)
• AI libraries (cuDNN, libDNN, ArmCL, OpenBLAS)

AI integration/services
• Cloud services (AWS, Google, Azure ...)

• Technological chaos: continuously changing algorithm/model/SW/HW stack


Docker images

VM images

• Non-representative / outdated training sets

• No common experimental frameworks and established methodologies which can adapt to this chaos

• Numerous reproducibility issues

• Very little artifact reuse in 1000+ ML papers

• Very little tech. transfer from academia (toy examples and too many papers)

• Docker, Kubernetes and VM images hide the mess but do not solve above problems

Public outcry about reproducibility and reusability crisis

Helping the society

Healthcare
Agriculture
Finances
Automotive
Aerospace
Meteorology
Retail
Robotics
…
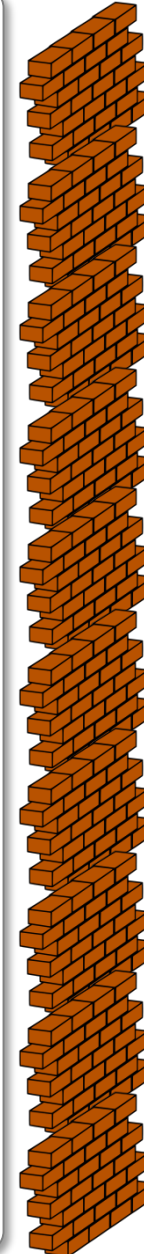
Many great tools, data sets and models to help researchers …

Applications
- Meteorology
- Health
- Robotics
- Automotive
- Economics
- Physics
- Astronomy
- Education

Scientific tools
- MATLAB
- Scilab
- Simulink
- LabVIEW
- Gnuplot
- LaTeX
- Ipython

Build tools
- Make
- Cmake
- SCons
- Bazel
- Gradle
- Ninja

Languages
- C++
- C#
- C
- Go
- PHP
- Fortran
- Java
- Python

Compilers
- LLVM
- GCC
- Intel
- PGI
- TVM
- CUDA

DevOps tools
- Git
- Jenkins
- Docker
- Kubernetes
- Singularity

Package managers
- Anaconda
- Go
- Npm
- Pip
- Sbt
- dpkg
- Spack
- EasyBuild

Libraries
- SciPy
- TFLite
- OpenBLAS
- MAGMA
- cuDNN
- cuFFT
- ArmNN
- CLBlast
- gemmlowp
- Boost
- HDF5
- MPI
- OpenCV
- Protobuf

OS
- Linux
- MacOS
- BSD
- Windows
- Android

Shells
- bash
- sh
- csh
- ksh
- Windows shell

Programs
- Image classification
- Object detection
- Natural Language processing
- Text processing
- Video processing
- Personal assistant

Benchmarks
- SPEC
- EEMBC
- HPCG
- LINPACK
- cBench
- MLPerf

Hardware
- CPU
- GPU
- TPU / NN
- DSP
- FPGA
- Quantum
- Simulators
- Interconnects

AI/ML frameworks
- TensorFlow
- PyTorch
- MXNet
- Caffe
- MCT (CNTK)
- Keras
- Kubeflow
- AutoML
- SageMaker
- Apache Spark

Models
- GoogleNet
- AlexNet
- VGG
- ResNet
- MobileNets
- SSD
- SqueezeNet
- DeepSpeech

Datasets
- ImageNet
- KITTI
- COCO
- MiDataSets
- Human Cell Atlas
- 1000 Genomes
- Earth models
- OpenStreetMap

Workload managers
- MPI
- SLURM
- PBS
- FLUX

Web services
- GitHub
- GitLab
- BitBucket
- Travis
- JupyterHub
- Codelabs
- SageMaker

Databases / experiments
- MySQL
- PostgreSQL
- MongoDB
- CouchDB
- Text files
- JSON files
- XLS files

Knowledge sharing
- ArXiv
- ACM DL
- IEEE DL
- GitHub
- Zenodo
- FigShare
- Web pages

Platforms
- HPC
- Desktops
- IoT
- Mobile
- Cloud sevices
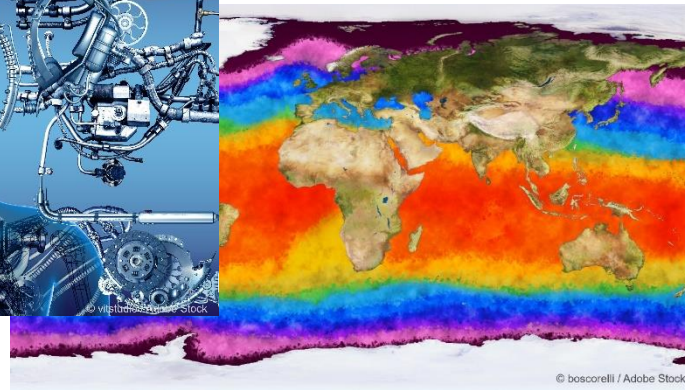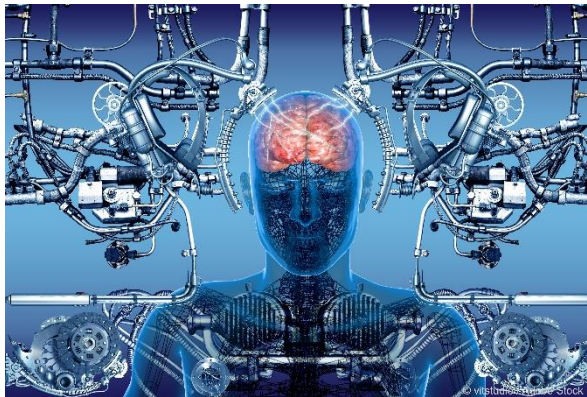
… but it's not easy to connect them together into reproducible AI / ML workflows

Applications
- Meteorology
- Health
- Robotics
- Automotive
- Economics
- Physics
- Astronomy
- Education

Programs
- Image classification
- Object detection
- Natural Language processing
- Text processing
- Video processing
- Personal assistant

AI/ML frameworks
- TensorFlow
- PyTorch
- MXNet
- Caffe
- MCT (CNTK)
- Keras
- Kubeflow
- AutoML
- SageMaker
- Apache Spark

Scientific tools
- MATLAB
- Scilab
- Simulink
- LabVIEW
- Gnuplot
- LaTeX
- Ipython

Models
- GoogleNet
- AlexNet
- VGG
- ResNet
- MobileNets
- SSD
- SqueezeNet
- DeepSpeech

Build tools
- Make
- Cmake
- SCons

- MiDataSets
- Human Cell Atlas
- 1000 Genomes
- Earth models
- OpenStreetMap

Languages
- C++
- C#
- C
- Go

managers
- MPI
- SLURM
- PBS
- FLUX

Compilers
- LLVM
- GCC
- Intel
- PGI

- GitLab
- BitBucket
- Travis
- JupyterHub
- Codelabs
- SageMaker

DevOps tools
- Git
- Jenkins
- Docker

- PostgreSQL
- MongoDB
- CouchDB
- Text files
- JSON files
- XLS files

Package managers
- Anaconda
- Go
- Npm

Knowledge sharing
- ArXiv
- ACM DL
- IEEE DL
- GitHub
- Zenodo
- FigShare
- Web pages

Libraries
- SciPy
- TFLite
- OpenBLAS
- MAGMA
- cuDNN
- cuFFT
- ArmNN
- CLBlast
- gemmlowp
- Boost
- HDF5
- MPI
- OpenCV
- Protobuf

Benchmarks
- SPEC
- EEMBC
- HPCG
- LINPACK
- cBench
- MLPerf

OS
- Linux
- MacOS
- BSD
- Windows
- Android

Shells
- bash
- sh
- csh
- ksh
- Windows shell

Hardware
- CPU
- GPU
- TPU / NN
- DSP
- FPGA
- Quantum
- Simulators
- Interconnects

Platforms
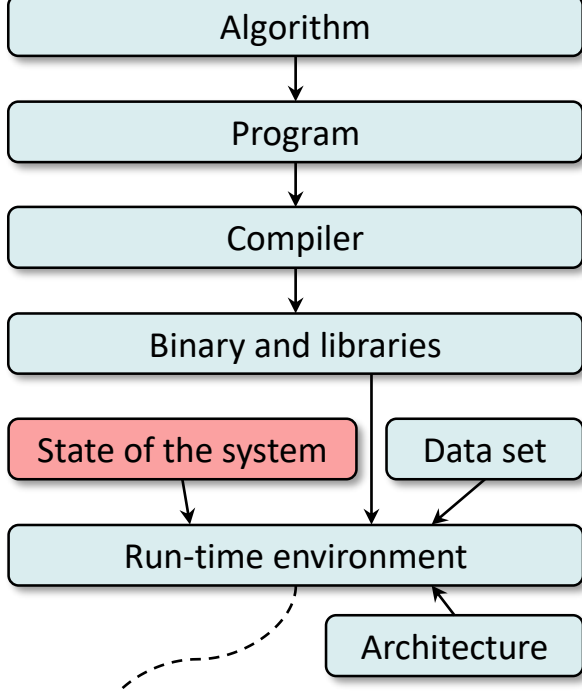- HPC
- Desktops
- IoT
- Mobile
- Cloud sevices

© Elnur / Adobe Stock

What I've noticed when reproducing papers at ACM and IEEE conferences since 2014

Authors develop their own ad-hoc scripts to do exactly the same "actions" across nearly all artifact submissions:

- Detect target hardware properties
- Detect software dependencies
- Install missing packages (code/data)
- Build code; run experiments
- Plot graphs and validate results
- Generate papers

Algorithm

Program

Compiler

Binary and libraries

State of the system | Data set

Run-time environment

Architecture

Result

Ad-hoc scripts to compile and run a program or a benchmark

image corner detection

matmul OpenCL

compression

neural network CUDA

Have some common meta: which datasets can use, how to compile, CMD, …

Ad-hoc scripts to install packages or set up environment for code and data deps on a given platform

GCC V8.1

LLVM V7.0

Intel Compilers 2017

Have some common meta: compilation, linking and optimization flags

Ad-hoc dirs for data sets with some ad-hoc scripts to find them, extract features, etc

image-jpeg-0001

bzip2-0006

txt-0012

video-raw-1280x1024

Have some (common) meta: filename, size, width, height, colors, …

Ad-hoc dirs and scripts to record and analyze experiments

cvs speedups

txt hardware counters

xls table with graph

Have some common meta: features, characteristics, optimizations

cTuning.org/ae

Authors develop their own ad-hoc scripts to do exactly the same "actions" across nearly all artifact submissions:

- Detect ta
- Detect s
- Install mi
- Build cod
- Plot and
- Generate

Ad-hoc scripts to compile and

image corner detection

Have some common meta:

State

Result

and analyze experiments

xls table with graph

characteristics, optimizations

The reason I started developing the open-source Collective Knowledge framework (CK) was

to help researchers share their artifacts
(code, data sets, models, scripts, experiments, papers)
as reusable, portable and customizable packages and workflows
with a simple Python API, CLI and JSON meta description.

I needed CK to support reproducibility initiatives
and help the community crowd-benchmark AI/ML techniques
across diverse software, hardware, data sets and models,
collaboratively reproduce and compare results,
and help companies move new techniques to production
while supporting any technology and legacy code

cKnowledge.org

CK concept : create, share and reuse ML/system "actions" as Python API, CLI and JSON
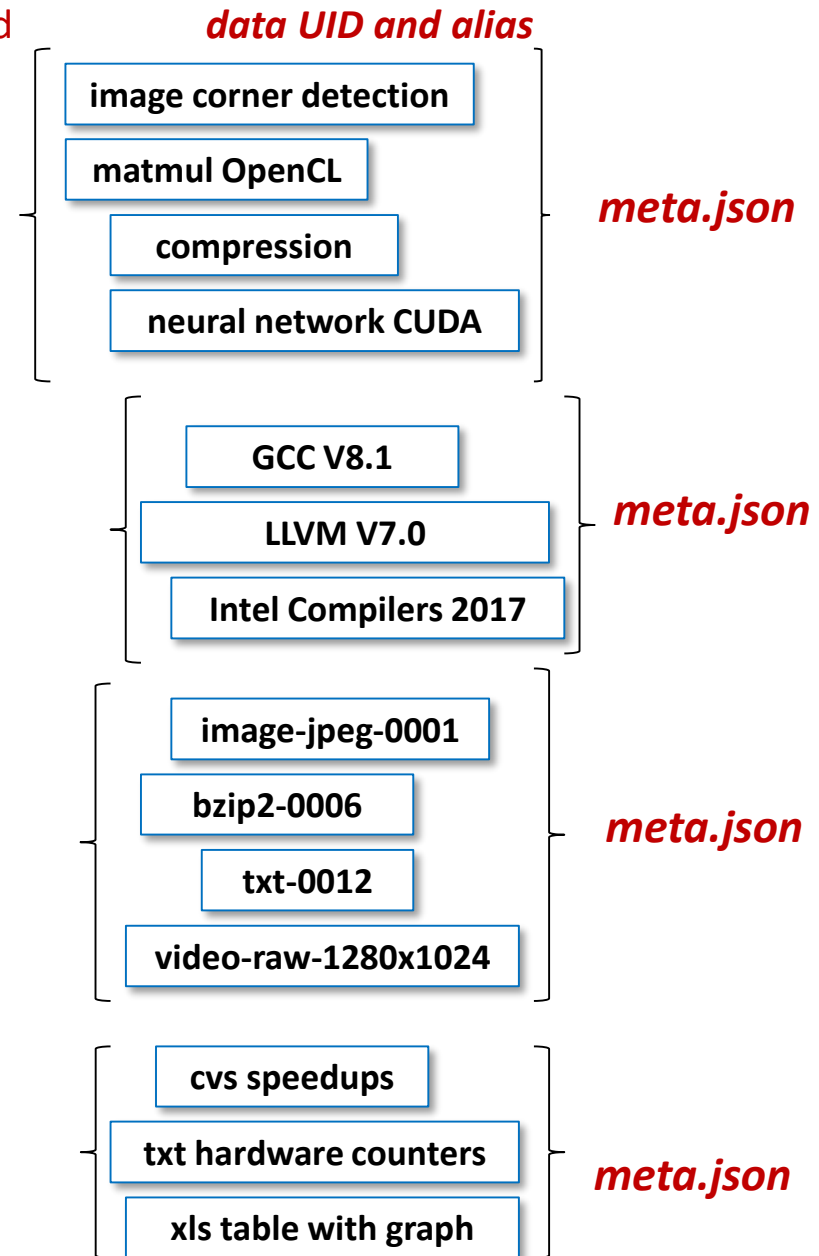
Such approach helped us bring standard DevOps practices and Continuous Integration to research and experimentation!

data UID and alias

Python module
"program"
with functions:
compile and run

- image corner detection
- matmul OpenCL
- compression
- neural network CUDA

meta.json

Python module
"soft"
with function:
detect

- GCC V8.1
- LLVM V7.0
- Intel Compilers 2017

meta.json

Python module
"dataset"
with function:
extract_features

- image-jpeg-0001
- bzip2-0006
- txt-0012
- video-raw-1280x1024

meta.json

Python module
"experiment"
with function:
add, get, analyze

- cvs speedups
- txt hardware counters
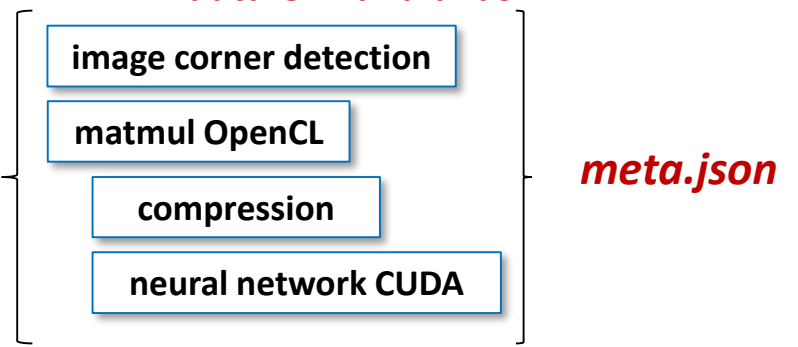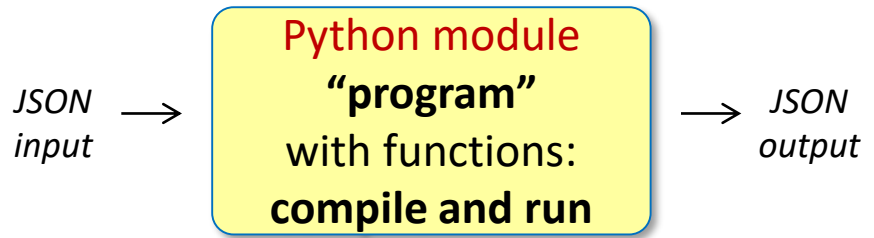- xls table with graph

meta.json

CK framework: simple CLI to create and access APIs (very portable - minimal dependencies)

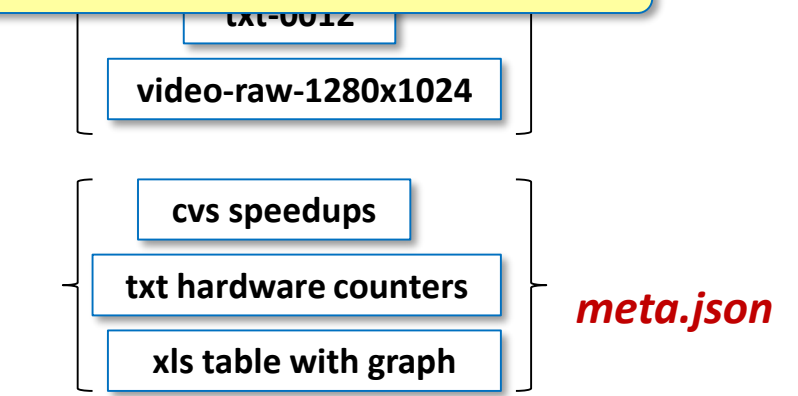CK: small python module (~200Kb); any python and git; Linux; Win; MacOS

$ pip install ck

$ ck {function} {module name}:{data name} @input.json

data UID and alias

JSON input → **Python module "program" with functions: compile and run** → *JSON output*

- image corner detection
- matmul OpenCL
- compression
- neural network CUDA

meta.json

JSON input → **Python module "soft" with function: detect** → *JSON output*

- GCC V8.1
- LLVM V7.0
- Intel Compilers 2017

meta.json

JSON input → **Python module "dataset" with function: extract_features** → *JSON output*

- image-jpeg-0001
- bzip2-0006
- txt-0012
- video-raw-1280x1024

meta.json

JSON input → **Python module "experiment" with function: add, get, analyze** → *JSON output*

- cvs speedups
- txt hardware counters
- xls table with graph

meta.json

CK framework: simple CLI to create and access APIs (very portable - minimal dependencies)

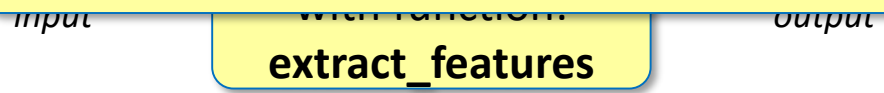CK: small python module (~200Kb); any python and git; Linux; Win; MacOS

$ pip install ck

$ ck {function} {module name}:{data name} @input.json
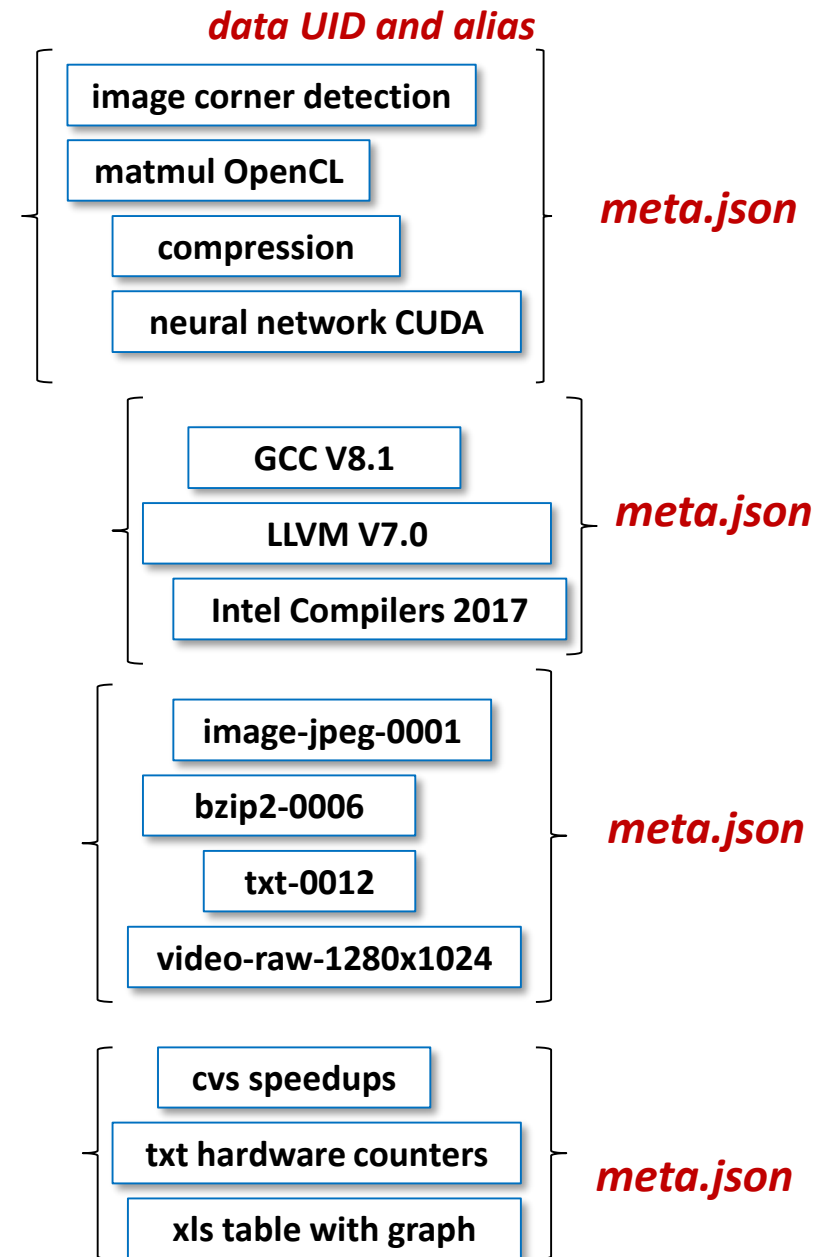
data UID and alias

JSON input → Python module **"program"** with functions: **compile and run** → *JSON output*

- image corner detection
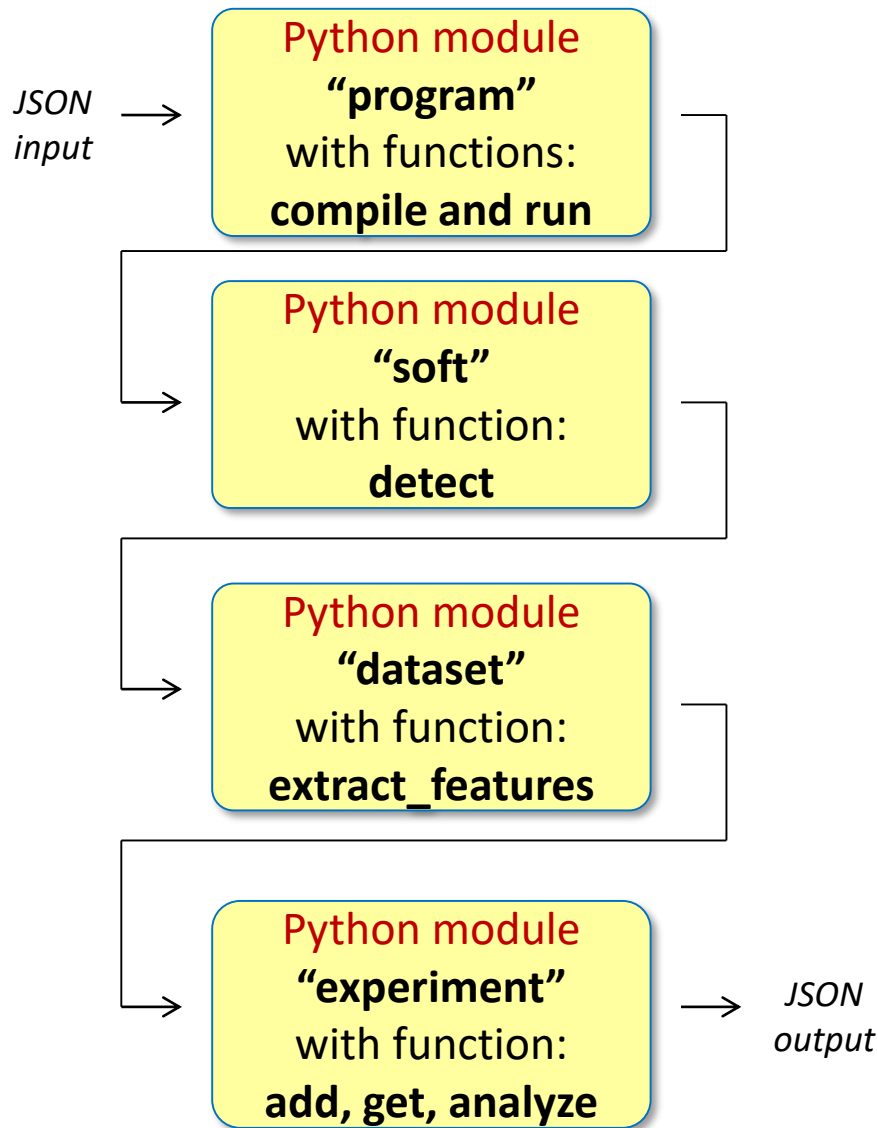- matmul OpenCL
- compression
- neural network CUDA

meta.json

600+ shared open-source automation actions from the past ACM, IEEE and ArXiv papers:

CodeReef.ai/portal/c/cr-action

ta.json

input → with function: **extract_features** → *output*

- txt-0012
- video-raw-1280x1024

ta.json

JSON input → Python module **"experiment"** with function: **add, get, analyze** → *JSON output*

- cvs speedups
- txt hardware counters
- xls table with graph

meta.json

CK concept: assemble customizable workflows with JSON input and output

CK workflows describe dependencies on other CK packages
using simple tags (code, data sets, models, frameworks, etc)

data UID and alias

JSON input →

Python module
"program"
with functions:
compile and run

Python module
"soft"
with function:
detect

Python module
"dataset"
with function:
extract_features

Python module
"experiment"
with function:
add, get, analyze

→ *JSON output*

image corner detection

matmul OpenCL

compression

neural network CUDA

meta.json

GCC V8.1

LLVM V7.0

Intel Compilers 2017

meta.json

image-jpeg-0001

bzip2-0006

txt-0012

video-raw-1280x1024

meta.json

cvs speedups

txt hardware counters

xls table with graph

meta.json

CK concept: provide simple and unified directory structure for shared artifacts

CodeReef.ai/portal/c/cr-repo

Collective Knowledge | COMPATIBLE

/ 1st level directory – CK modules

setup	soft
find	dataset
extract features	
compile	program
run	
autotune	
add	experiment
replay	

JSON API   Python module

/ 2nd level dir - CK entries

TensorFlow
PyTorch
ARM compute lib
ImageNet
Car video stream
Real surveillance camera
image classification
object detection
GEMM OpenCL
convolution CPU
performance results
training / accuracy
bugs

holder for original artifact

/ CK meta info

JSON file
JSON file
JSON file
JSON file
JSON file
JSON file
JSON file
JSON file
JSON file
JSON file
JSON file
JSON file
JSON file

CK meta

We started developing CK components to solve AI / ML portability and adaptation to a given environment

1) Describing different operating systems

```
$ ck pull repo:ck-env
$ ck ls os
$ ck load os:linux-64 --min
```

Needed to support real use cases from our partners:
https://cKnowledge.org/partners

2) Detecting and unifying information about platforms

```
$ ck detect platform --help
$ ck detect platform --out=json
$ ck load os:linux-64 --min
```

3) Detecting installed "software" (both code and data):

```
$ ck search soft --tags=dataset
$ ck detect soft:compiler.llvm

$ ck show env --tags=llvm
```

250+ software detection plugins:
codereef.ai/portal/c/soft

4) Installing missing packages (both code and data): front-end to EasyBuild, Spack, scons …

```
$ ck search package --tags=model
$ ck install compiler:compiler-llvm-7.0.0-universal
$ ck show env --tags=llvm
$ ck virtual env –tags=llvm,v7.0.0
```

600+ shared packages:
codereef.ai/portal/c/package

We can now adapt to any native environment and use containers to make stable snapshots

github.com/ctuning/ck/wiki/Portable-workflows

Artifact automated and reusable   Collective Knowledge COMPATIBLE   Workflow CK

We then started developing CK workflows for portable and customizable AI / ML benchmarking

Common JSON API

Algorithm / source code

JSON meta

AI framework

Available libraries / skeletons

Compilers

Binary or byte code

Inputs		Various models

Run-time environment

Run-time state of the system		Hardware, simulators

We developed a universal program workflow to compile, run, profile and autotune AI/ML applications across diverse models, data sets and platforms, validate results, record experiments, share and reproduce them, and report discrepancies

```
$ ck pull repo:ck-crowdtuning

$ ck ls program
$ ck ls dataset

$ ck load program:cbench-automotive-susan  --min
$ ck compile program:cbench-automotive-susan –fast

$ ck run program:cbench-automotive-susan

$ ck autotune program:cbench-automotive-susan

$ ck crowdtune program:cbench-automotive-susan

$ ck replay experiment
```
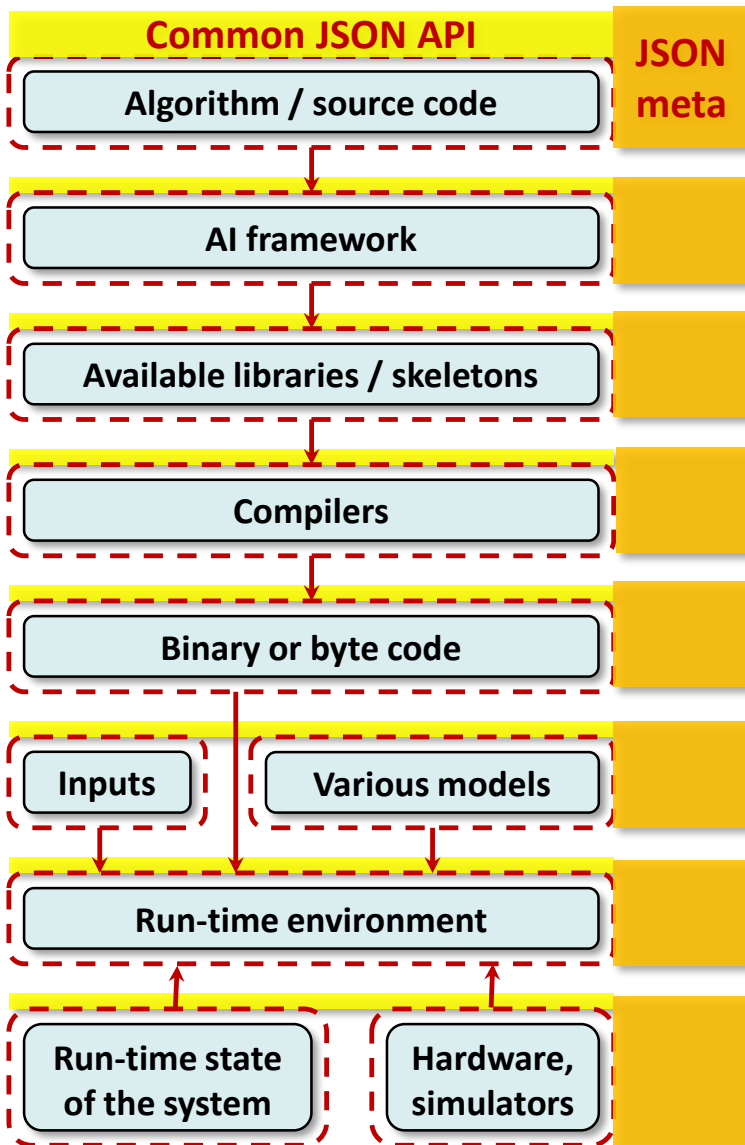
CK workflows describe dependencies on CK soft detection plugins and packages to automatically adapt to a given platform and environment

CodeReef.ai/portal/c/cr-solution/demo-obj-detection-coco-tf-cpu-benchmark-linux-portable-workflows/#dependencies

Customizable CK workflows can be also used to autotune the whole AI/ML/SW/HW stack!

First expose <u>coarse grain high-level</u> choices, features, system state and behavior characteristics via CK APIs

CK Python modules (wrappers) with a unified JSON API

CK input (JSON/dict)

i

Unified input

Action

Behavior

Choices

Features

State

Some actions

Set environment for a given tool version

Unified output

Behavior

Choices

Features

State

CK output (JSON/dict)

i

Tools (compilers, profilers, etc) → Generated files

Formalized function B of a behavior of any CK object

$$\vec{b} = B(\vec{c}, \vec{f}, \vec{s})$$

Flattened CK JSON vectors
(dict converted to vector)
to simplify statistical analysis, machine learning and data mining

Then automate crowd-benchmarking and optimization across diverse models, datasets and platforms

Chain CK modules to implement research workflows such as multi-objective autotuning and co-design

i

| Choose exploration strategy | Perform SW/HW DSE (math transforms, skeleton params, compiler flags, transformations …) | *CK program module with pipeline function* Compile program → Run code | Perform stat. analysis | Detect (Pareto) frontier | Model behavior, predict optimizations | Reduce complexity |

i
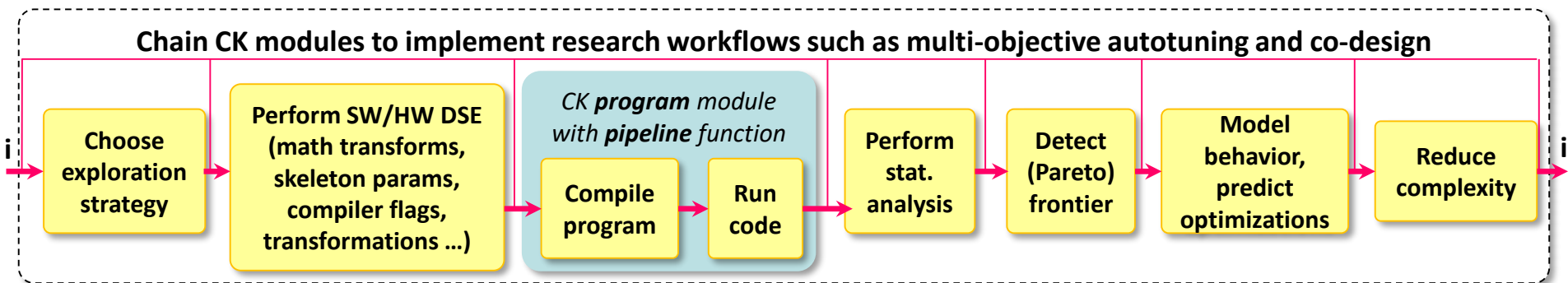
Keep best species (AI/SW/HW choices); model behavior; predict better optimizations and designs
CodeReef.ai/portal/c/cr-result

Gradually expose more optimizations and characteristics via JSON files

Autotuning and machine learning specification:

CK flattened JSON key

##characteristics#execution_times@1

```
{

  "characteristics":{
    "execution times": ["10.3","10.1","13.3"],
    "code size": "131938", ...},
  "choices":{
    "os":"linux", "os version":"2.6.32-5-amd64",
    "compiler":"gcc", "compiler version":"4.6.3",
    "compiler_flags":"-O3 -fno-if-conversion",
    "platform":{"processor":"intel xeon e5520",
          "l2":"8192", ...}, ...},
  "features":{
    "semantic features": {"number_of_bb": "24", ...},
    "hardware counters": {"cpi": "1.4" ...}, ... }
  "state":{
    "frequency":"2.27", ...}

}
```

```
"flattened_json_key":{
    "type": "text"|"integer" | "float" | "dict" | "list" | "uid",
    "characteristic": "yes" | "no",
    "feature": "yes" | "no",
    "state": "yes" | "no",
    "has_choice": "yes" | "no",
    "choices": [ list of strings if categorical choice],
    "explore_start": "start number if numerical range",
    "explore_stop": "stop number if numerical range",
    "explore_step": "step if numerical range",
    "can_be_omitted" : "yes" | "no"
    ...
}
```

AI hardware
• All major vendors (Google, NVIDIA, ARM, Intel, IBM, Qualcomm, Apple, AMD …)

AI models
Many groups in academia & industry (Google, OpenAI, Microsoft, Facebook …)

AI software
• AI frameworks (TensorFlow, MXNet, PyTorch, CNTK, Theano)
• AI libraries (cuDNN, libDNN, ArmCL, OpenBLAS)

AI integration/services
• Cloud services (AWS, Google, Azure ...)

cKnowledge.org/request
Finding the most efficient AI/SW/HW stacks across diverse models, data sets and platforms via open competitions, share them as reusable CK components and visualize on a public scoreboard

Organizers (A-Z)

Luis Ceze, University of Washington
Natalie Enright Jerger, University of Toronto
Babak Falsafi, EPFL
Grigori Fursin, cTuning foundation
Anton Lokhmotov, dividiti
Thierry Moreau, University of Washington
Adrian Sampson, Cornell University
Phillip Stanley Marbell, University of Cambridge

Real use-cases

Healthcare
Agriculture
Finances
Automotive
Aerospace
Meteorology
Retail
Robotics
…

CK platform & CodeReef
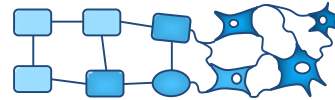
Interdisciplinary community

We reproduced results from published ML papers and shared portable ML workflows!

8 intentions to submit and 5 submitted image classification workflows with unified Artifact Appendices

TensorFlow; Keras; Avro
↓
AlexNet, VGG16
↓
Nvidia Jetson TX2; Raspberry Pi with ARM

Intel Caffe ; BVLC Caffe
↓
ResNet-50; Inception-V3; SSD 32-bit ; 8-bit
↓
Intel C++ Compiler 17.0.5 20170817
↓
AWS; Xeon® Platinum 8124M

MXNet; NNVM/TVM
↓
OpenBLAS vs ArmCL
↓
VGG16, MobileNet and ResNet-18
↓
GCC; LLVM
↓
Firefly-RK3399

MXNet; NNVM/TVM
↓
ResNet-*
↓
Xilinx FGPA (Pynq board)

ArmCL 18.01 vs 18.02 vs dividiti (OpenCL)
↓
MobileNets
↓
GCC
↓
HiKey 960 (GPU)

Public validation at github.com/ctuning/ck-request-asplos18-results via GitHub issues.

All validated papers are published in the ACM DL
with **portable, customizable and reusable CK components and workflows**:
dl.acm.org/citation.cfm?doid=3229762

See ACM ReQuEST report: portalparts.acm.org/3230000/3229762/fm/frontmatter.pdf

Multi-objective results for all AI/SW/HW stacks are presented on a live scoreboard and become available for public comparison and further customization, optimization and reuse!

We are not announcing a single winner! We show all multi-dimensional results at CodeReef.ai/portal/c/cr-result/pareto-efficient-ai-co-design-tournament-request-acm-asplos-2018 and let the users select best ML/SW/HW stacks depending on multiple constraints for their production use!

Multi-objective results for all AI/SW/HW stacks are presented on a live scoreboard and become available for public comparison and further customization, optimization and reuse!
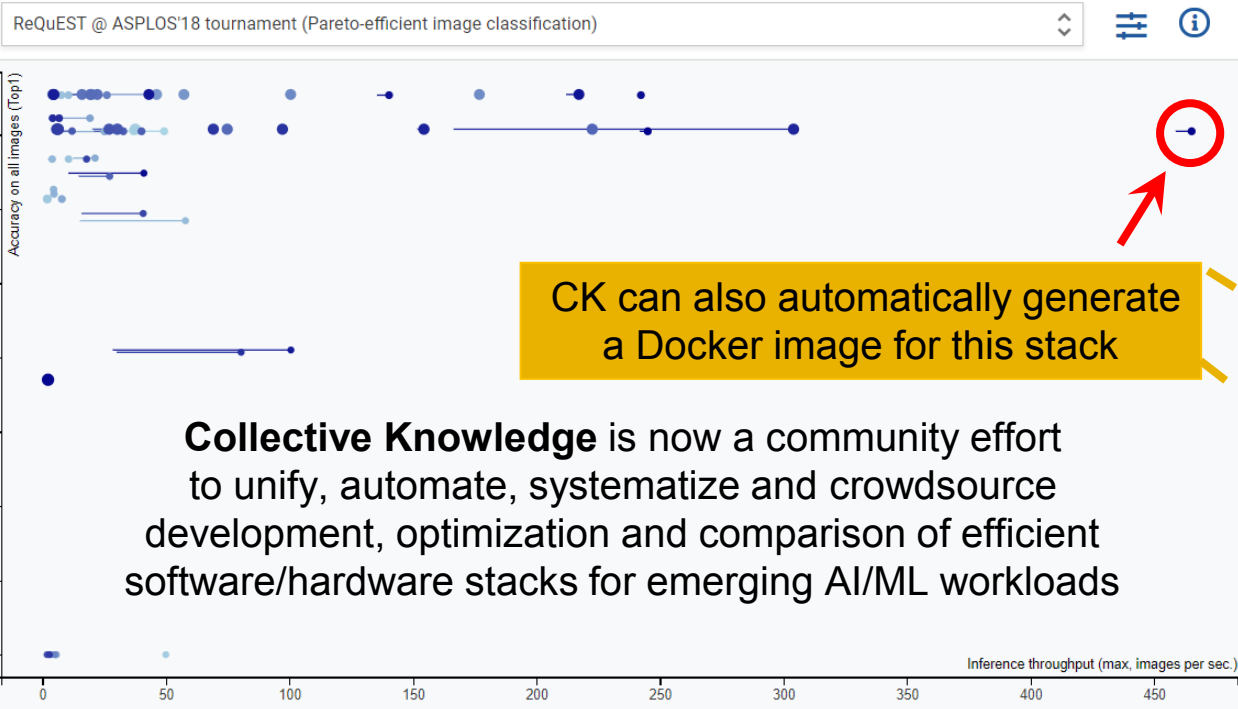
CK assists AWS market place with collaboratively optimized AI/ML stacks

CK can also automatically generate a Docker image for this stack

Collective Knowledge is now a community effort to unify, automate, systematize and crowdsource development, optimization and comparison of efficient software/hardware stacks for emerging AI/ML workloads

Accelerate technology transfer: companies can validate published techniques in their production environment using shared CK workflows!

We made a joint presentation with Amazon at O'Reilly AI conference (October 2018)

General Motors uses CK to select the most efficient SW/HW stacks

Performance, accuracy, power consumption practically never match official reports!

CK workflows and automation helped GM evaluate numerous models, datasets, frameworks and libraries to find the most efficient SW/HW stacks for object detection across Nvidia, AMD, ARM and Intel platforms (CUDA, OpenCL, OpenMP …)

GM presentation about using CK: www.youtube.com/watch?v=1ldgVZ64hEI

Continuously collect statistics, bugs and misclassifications
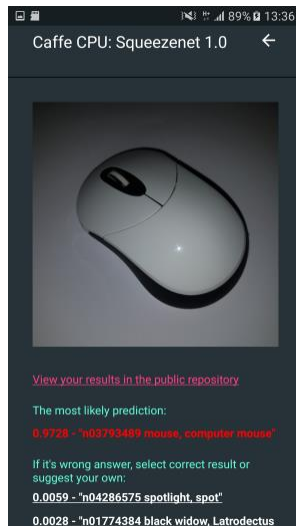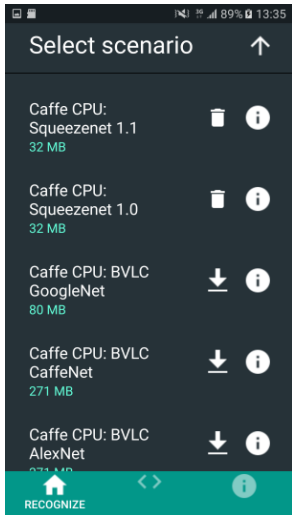
The number of distinct participated platforms: **800+**
The number of distinct CPUs: **260+**
The number of distinct GPUs: **110+**
The number of distinct OS: **280+**
Power range: **1-10W**

No need for a dedicated and expensive cloud – volunteers help us validate research ideas similar to SETI@HOME

Also collecting real images from users for misclassifications to build an open and continuously updated training set)!

Firefly-RK3399

Better accuracy

Winning solutions on various frontiers

Time per image (seconds)
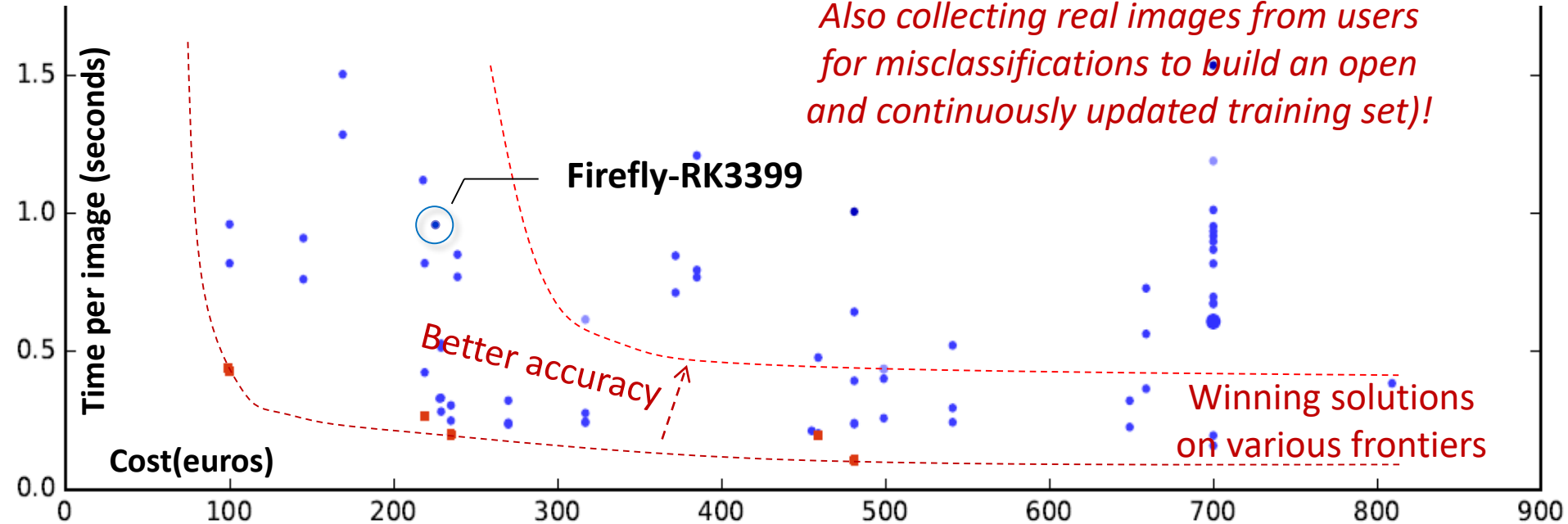
Cost(euros)

"**MobileNets**: Efficient Convolutional Neural Networks for Mobile Vision Applications" (Andrew G. Howard et al., 2017, https://arxiv.org/abs/1704.04861):

- Parameterised CNN family using depthwise separable convolutions.
- Channel multiplier: 1.00, 0.75, 0.50, 0.25 - marker shape (see below).
- Input image resolution: 224, 192, 160, 128 - marker size.

CodeReef.ai/portal/search/?q="SOTA+MLPerf"

A broad ML benchmark suite for measuring performance of ML software frameworks, ML hardware accelerators, and ML cloud platforms.

It is even possible to test how object detection from MLPerf works live:
CodeReef.ai/portal/c/cr-solution/demo-obj-detection-coco-tf-cpu-webcam-linux-azure

Use CK to autotune MobileNets across diverse devices for MLPerf submissions

The MLPerf consortium has recently released over 500 validated inference benchmarking results from 14 organizations (including DellEMC, Nvidia, Google, Intel, Alibaba, Habana) measuring how fast and how well a pre-trained computer system can classify images, detect objects, and translate sentences.

Over 400 of these results were automated with the CK framework and CK workflows.
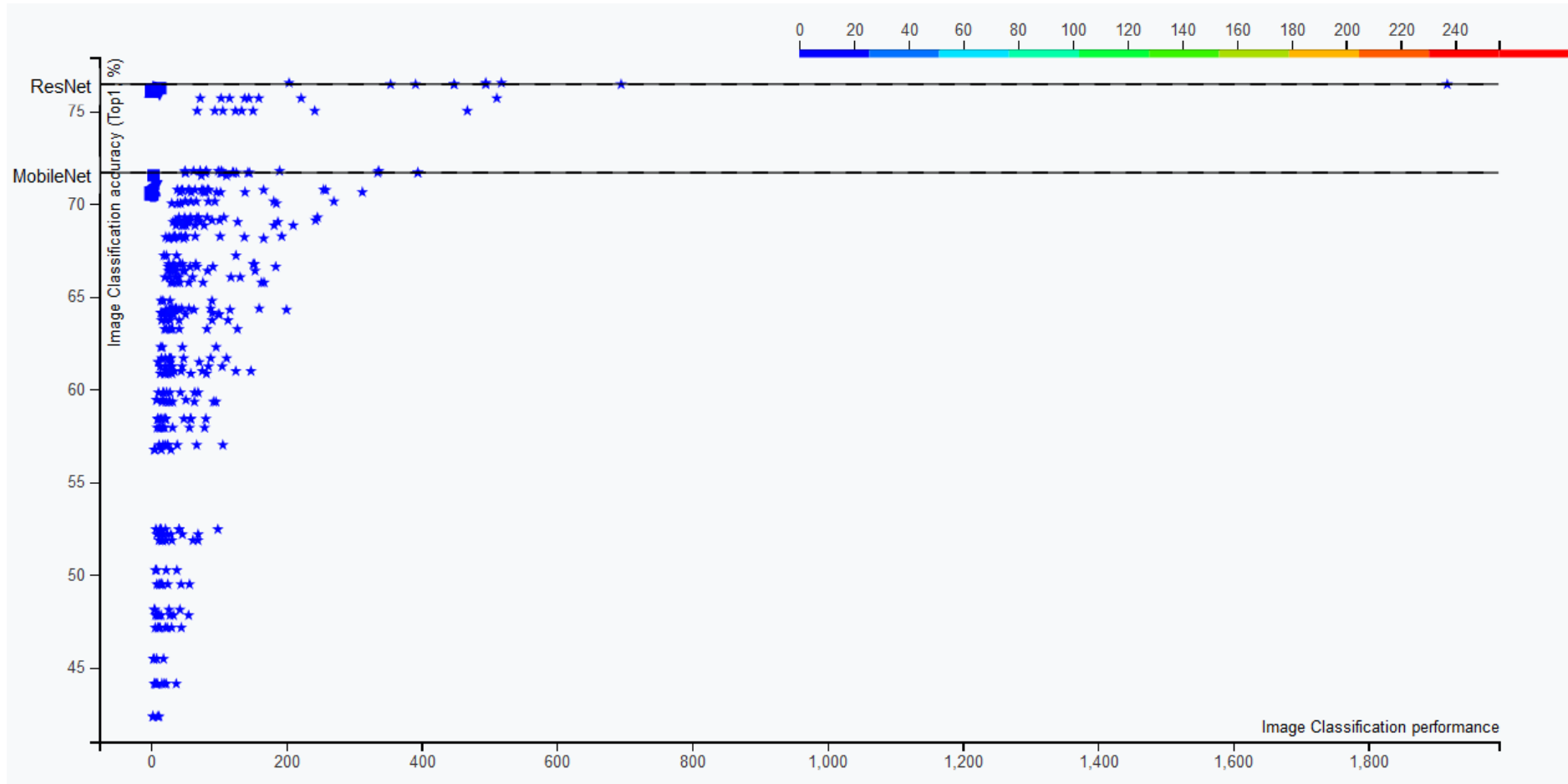
cKnowledge.org/dashboard   mlperf.org

- CK is a promising open-source technology already used by companies and universities in production but there is still a lot to be improved!

  downloads 140k | pypi package 1.12.2 | python 2.7 | 3.4+ | License BSD 3-Clause

- Current major issues preventing further adoption:
    - CLI and JSON meta is not user friendly (similar to Git)
    - Distributed nature of CK makes it difficult to understand who is using CK and ensure the stability/testing of workflows
    - Lack of an open portal to exchange stable components (similar to PyPi)
    - Lack of automatic testing of all components and workflows

- Currently supported by my non-profit cTuning foundation (cTuning.org) but our resources are very limited

- Want to attract more companies to improve our open-source technology together

Based on the feedback from the CK users I an now working
on an open and free portal to solve above problems,
continue supporting our open-source developments,
and enable practical and portable MLOps
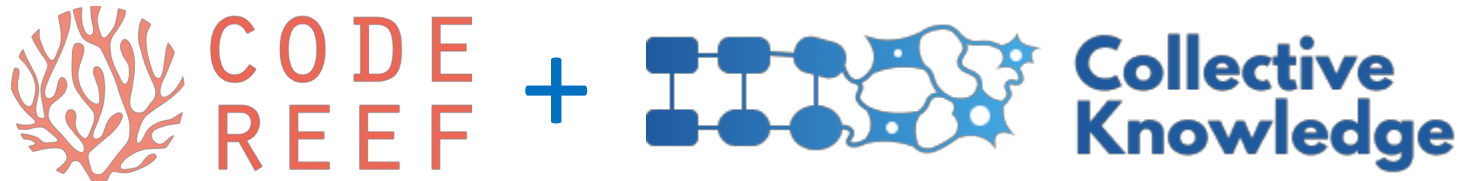
CodeReef.ai/portal
CodeReef.ai/static/docs

Reproduced papers with portable CK workflows:
CodeReef.ai/portal/search/?q="portable-workflow-ck"

We will present CodeReef+CK for MLOps at the following events:
- NeurIPS MLSys'20 (Austin), mlsys.org
- AI hardware summit (Munich),
  www.kisacoresearch.com/events/ai-hardware-summit-europe
- ACM ASPLOS (Lausanne), asplos-conference.org
- IEEE ISPASS (Boston), www.ispass.org/ispass2020

CodeReef.ai: an open platform for portable AI/ML benchmarking and MLOps

Repository of customizable, portable and reusable research components with CK API

CK JSON API
AI frameworks
- TensorFlow
- Caffe
- Caffe2
- CNTK
- Torch
- MXNet
- ...

CK JSON API
Models
- AlexNet
- GoogleNet
- VGG
- ResNet
- SqueezeDet
- SSD
- MobileNets
- ...

CK JSON API
Libraries
- cuDNN
- ArmCL
- OpenBLAS
- ViennaCL
- CLBlast
- cuBLAS
- TVM
- ...

CK JSON API
Data sets
- KITTI
- COCO
- VOC
- ImageNet
- Real life objects from the community
- ...

CK JSON API
OS
| Linux | Windows |
| MacOS | Android |
...

CK JSON API
Hardware
CPU	GPU
DSP	NN accelerators
FPGA	Simulators
...

Customizable CK workflows
for real-world AI/ML tasks

Assemble scenarios such as image classification as LEGO™

CK JSON API
Models
▢ ▢ ▢

CK JSON API
Software
▢ ▢ ▢

CK JSON API
Data sets
▢ ▢ ▢

CK JSON API
Hardware
▢ ▢ ▢

Share complete workflows along with published papers to automate artifact evaluation
and help the community build upon prior work

Crowdsource experiments, benchmarking and co-design with the help of volunteers
across diverse models, data sets and platforms

Present best results, AI/ML workflows and components
on a live scoreboard for fair comparison and reuse

Help researchers learn ML/AI techniques,
quickly prototype new ones,
validate them in practice with companies,
and even contribute back new research components

Collaboratively benchmark and co-design
SW/HW stacks, select the most efficient AI/ML solutions,
reduce development costs, and increase productivity

CodeReef.ai/portal

Contact

Grigori.Fursin@cTuning.org or Grigori.Fursin@CodeReef.ai

LF AI Updates

Outreach Committee

Recent Announcements

1. **New Member Welcome - RStudio + inwinSTACK + ISSIP:** https://lfai.foundation/blog/2020/01/30/lf-ai-foundation-new-member-welcome-rstudio-inwinstack-issip/
2. **Sparklyr New Project:** https://lfai.foundation/blog/2020/01/29/sparklyr-joins-lf-ai-as-its-newest-incubation-project/
3. **LF AI 2019 Year in Review**: https://lfai.foundation/blog/2020/01/22/lf-ai-2019-year-in-review/
4. **Horovod 0.19.0 Release:** https://lfai.foundation/blog/2020/01/14/horovod-version-0-19-0-now-available/
5. **Angel Graduation:** https://lfai.foundation/blog/2019/12/19/lf-ai-foundation-announces-graduation-of-angel-project/
6. **Zilliz new premier member:** https://lfai.foundation/blog/2019/12/17/lf-ai-welcomes-zilliz/
7. **LF AI Day Shanghai Summary:** https://lfai.foundation/blog/2019/12/05/thank-you-lf-ai-day-shanghai-summary/
8. **LF AI Receives Contribution Award from CAAI:** https://lfai.foundation/blog/2019/11/27/lf-ai-receives-best-contribution-award/
9. **Acumos Clio Release:** https://lfai.foundation/press-release/2019/11/26/lf-ai-delivers-acumos-ai-clio-release/
10. **Pyro 1.0.0 Release:** https://lfai.foundation/blog/2019/11/18/pyro-1-0-has-arrived/

Upcoming Announcements

March:

Acumos and Angel Collaboration
Marquez New Project
Milvus New Project
Angel Roadmap + Project Participation Invite
LF AI 2 Year Anniversary
ITU Global Challenge on AI/ML in 5G Networks Promotion

April:

Q2 New Member Announcements
IEEE Infrastructure Conference - LF AI Participation
Open Networking & Edge Summit

LF AI Events

Events Page on LF AI Website:
https://lfai.foundation/events/

Events Calendar on LF AI Wiki (subscribe available):
https://wiki.lfai.foundation/pages/viewpage.action?pageId=12091544

2020 Events Wiki:
https://wiki.lfai.foundation/display/DL/2020+Events

The Outreach Committee and LF AI Community need to identify Event Leads to drive these events as the Foundation is not resources/budgeted to manage them all. Please encourage participation within your organizations. LF AI provides event planning templates/resources and is available to help with guidance and questions along the way. Please email the Jessica Kim (Outreach Committee Chair) or Jacqueline/Ibrahim to discuss participation.

Call to Participate in Ongoing Efforts

Trusted AI

› **Leadership:**
Animesh Singh (IBM), Souad Ouali (Orange), and Jeff Cao (Tencent)
› **Goal:** Create policies, guidelines, tooling and use cases by industry
› **Github:**
https://github.com/lfai/trusted-ai
› **Wiki:**
https://wiki.lfai.foundation/display/DL/Trusted+AI+Committee
› **To participate:**
https://lists.lfai.foundation/g/trustedai-committee/
› **Next call:** Bi-weekly on Thursdays at 7am PT, subscribe to group calendar on wiki
https://wiki.lfai.foundation/pages/viewpage.action?pageId=12091895

ML Workflow

› **Leadership:**
Ofer Hermoni
› **Goal:**
Define an ML Workflow and promote cross project integration
› **Wiki:**
https://wiki.lfai.foundation/display/DL/ML+Workflow+Committee
› **To participate:**
https://lists.lfai.foundation/g/mlworkflow-committee
› **Next call:** Bi-weekly on Thursdays at 7:00 am PT, subscribe to group calendar on wiki
https://wiki.lfai.foundation/pages/viewpage.action?pageId=18481242

Upcoming TAC Meetings

Upcoming TAC Meetings

March 12: Incubation Project Proposal + TAC Vote: NNStreamer from Samsung

March 26: TBD

TAC Meeting Details

- › To subscribe to the TAC Group Calendar, visit the wiki: https://wiki.lfai.foundation/x/XQB2
- › Join from PC, Mac, Linux, iOS or Android: https://zoom.us/j/430697670

- › Or iPhone one-tap:
  - › US: +16465588656,,430697670# or +16699006833,,430697670#

- › Or Telephone:
  - › Dial(for higher quality, dial a number based on your current location):
  - › US: +1 646 558 8656 or +1 669 900 6833 or +1 855 880 1246 (Toll Free) or +1 877 369 0926 (Toll Free)
- › Meeting ID: 430 697 670

- › International numbers available: https://zoom.us/u/achYtcw7uN

THE **LINUX** FOUNDATION   **LF** AI

Open Discussion

THE **LINUX** FOUNDATION

LF AI

Legal Notices

THE LINUX FOUNDATION

LF AI