CK: an open-source framework to automate, reproduce, crowdsource and reuse experiments at HPC conferences

github.com/ctuning/ck cKnowledge.org/partners

Companies, universities, and non-profits do cool things with the help of CK

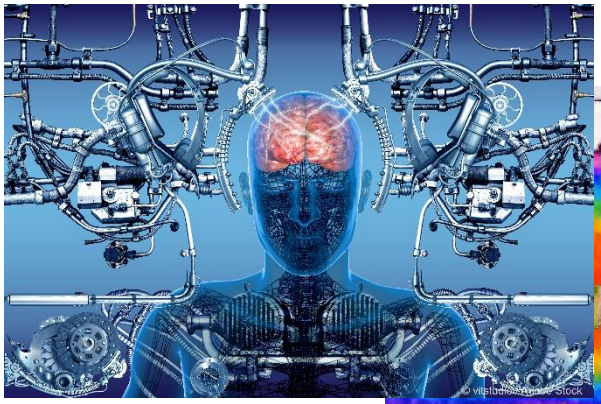
CK complements many popular tools and services

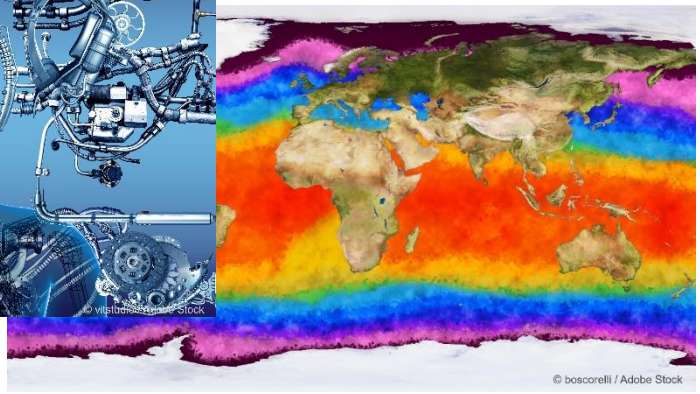
Grigori Fursin @grigori_fursin fursin.net/research

AI, ML, systems and quantum research is booming – 1000+ papers every year …

Applications

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





Platforms

- HPC
- Desktops
- IoT
- Mobile
- Cloud sevices

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

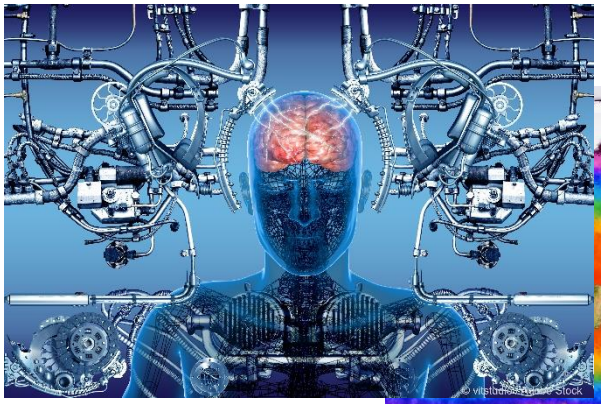
Package managers
- Anaconda
- Go
- Npm
- Pip
- Sbt
- dpkg
- Spack
- EasyBuild

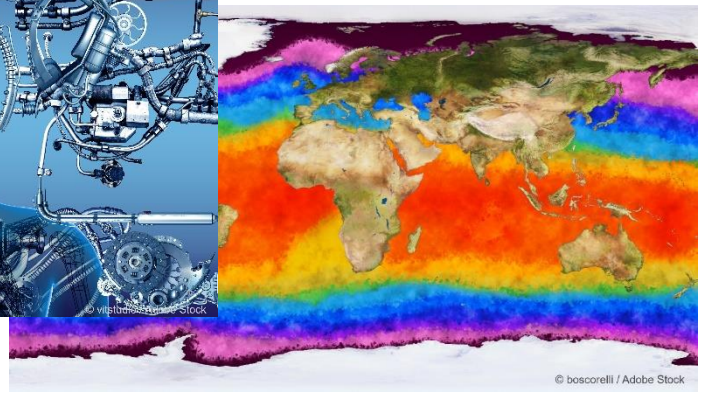
Libraries
- SciPy
- TFLite
- OpenBLAS
- MAGMA
- cuDNN
- cuFFT
- ArmNN
- CLBlast
- gemmlowp
- Boost
- HDF5
- MPI
- OpenCV
- Protobuf

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

Let's innovate …

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

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

Build tools
- Make
- Cmake
- SCons

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

Compilers
- LLVM
- GCC
- Intel
- PGI

DevOps tools
- Git
- Jenkins
- Docker

Package managers
- Anaconda
- Go
- Npm

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

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

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


© Elnur / Adobe Stock

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

managers
- MPI
- SLURM
- PBS
- FLUX

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

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

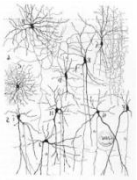
Benchmarks
- SPEC
- EEMBC
- HPCG
- LINPACK
- cBench
- MLPerf

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

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

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

Reproducing results at HPC conferences (artifact evaluation): the good, the bad, and the ugly



Idea

Creating ad-hoc project with many add-hoc scripts and README files, and sharing it via GitHub, GitLab, ...
doesn't make it easily portable, reusable and customizable!

- 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*

Very often software from published papers die when students leave or projects finish!

Current solution: let's just hide the mess …

Creating ad-hoc project with many add-hoc scripts and README files, and sharing it via GitHub, GitLab

Ad-hoc scripts to compile and

image corner detection

OpenCL

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

Virtual Machines
Containers
…

Result

and analyze experiments

txt hardware counters

xls table with graph

Have some common meta: *features, characteristics, optimizations*

Very often software from published papers die when students leave or projects finish!

Collective Knowledge concept (CK): add, share and reuse common APIs!

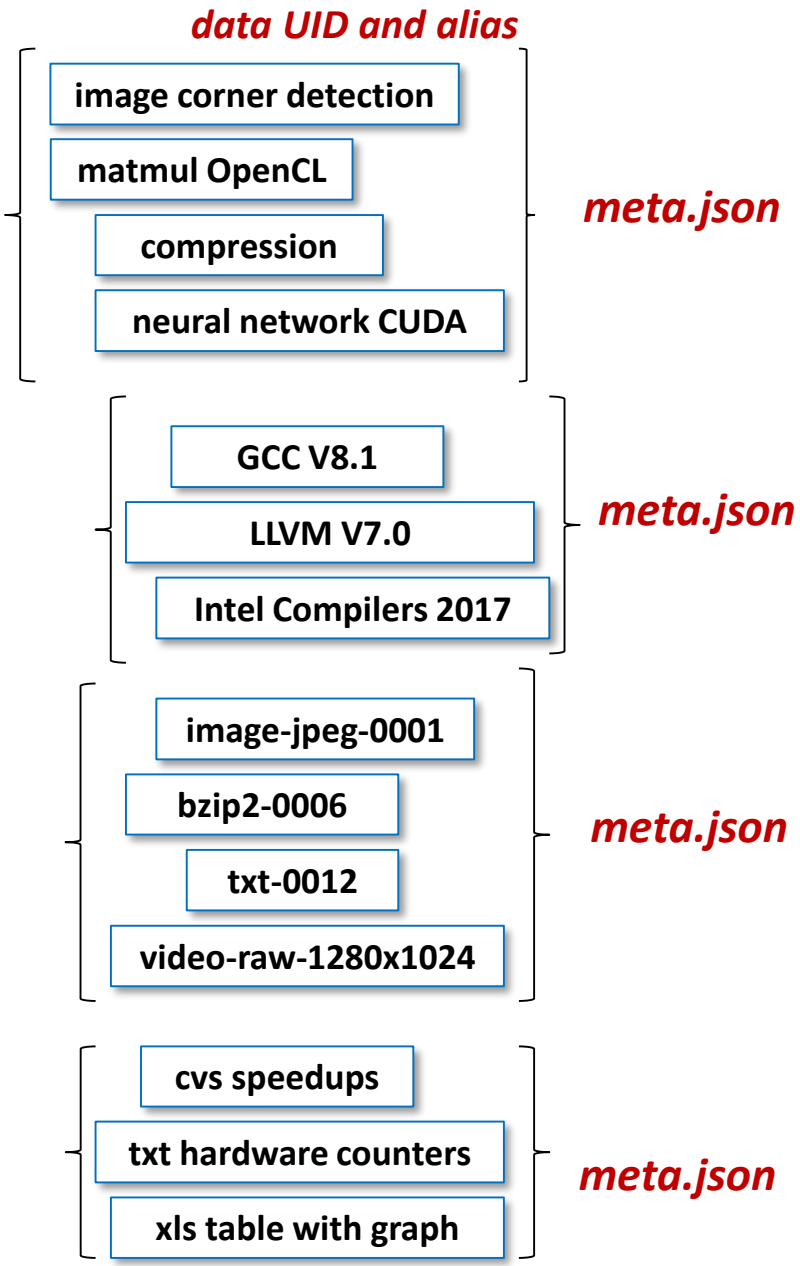
Provide unified Python APIs and JSON meta descriptions for similar code and data objects – enables DevOps!

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



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

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

$ 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

cKnowledge.org/shared-modules.html

CK framework: the community can implement complex workflows via CK APIs

CK: small python module (~200Kb); no extra dependencies; Linux; Win; MacOS

Assemble workflows from shared components

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*

data UID and alias

- 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

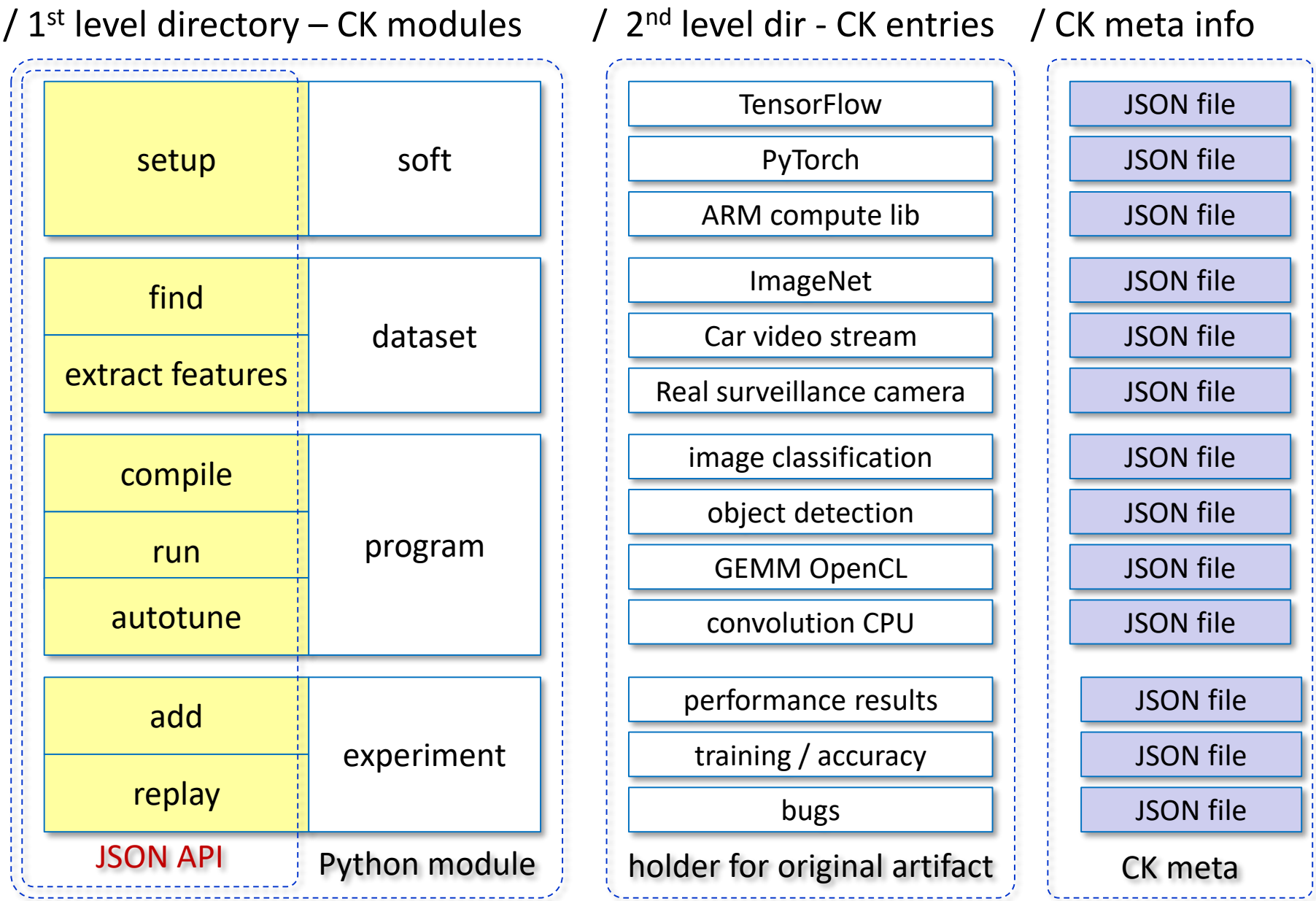
meta.json

cKnowledge.org/shared-programs.html

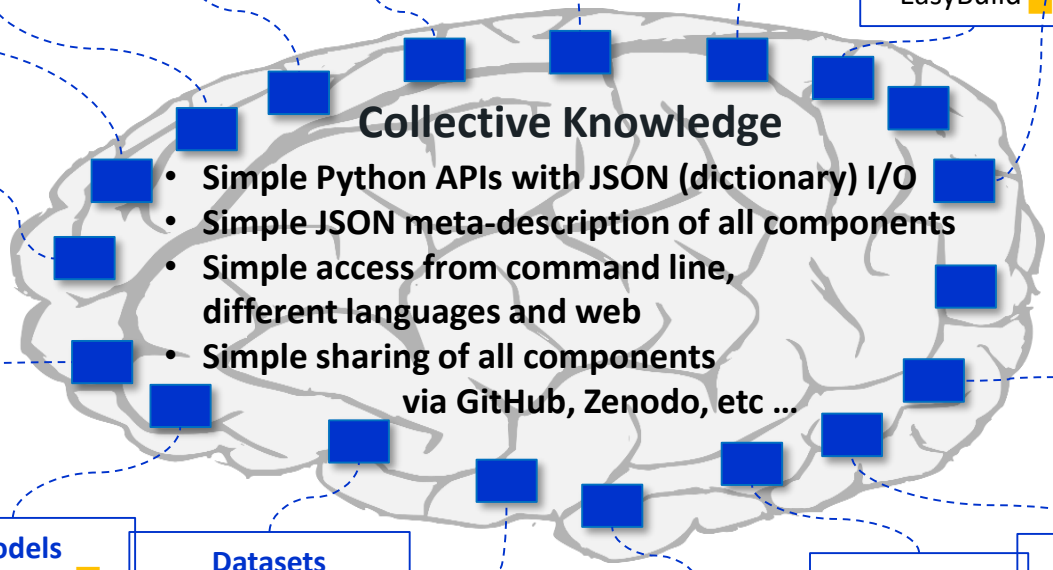
CK framework: provide simple and unified directory structure for research projects

cKnowledge.org/shared-repos.html

Collective Knowledge COMPATIBLE



We started collaboratively abstract all components with our partners since 2017 …



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

Started converting all my own past R&D on performance autotuning and machine learning to CK …

cKnowledge.org/shared-repos.html cKnowledge.org/shared-modules.html

1) **Describing different operating systems** (github.com/ctuning/ck-env)

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

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
```

cKnowledge.org/shared-soft-detection-plugins.html

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

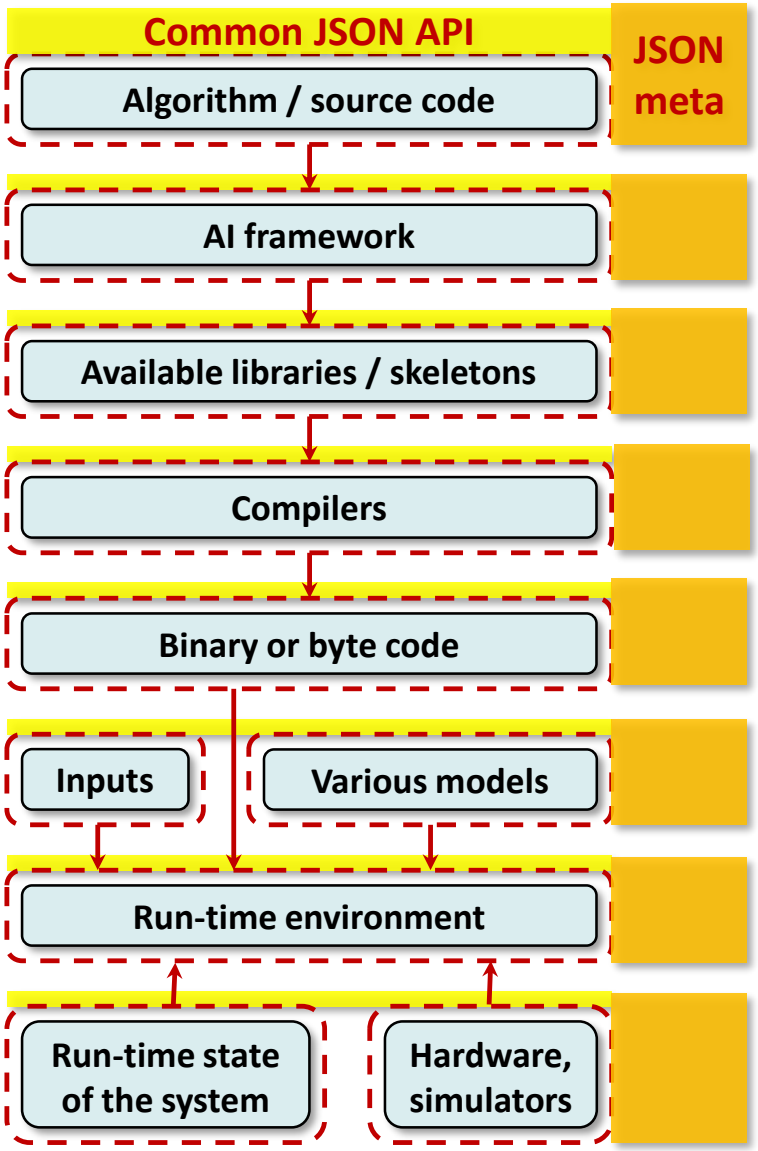
4) **Installing missing packages** (**both** code and data): front-end to EasyBuild, Spack, scons …

```
$ 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
```

cKnowledge.org/shared-packages.html

I want to work in a native environment and use containers to make stable snapshots

Enabling customizable and portable workflows by connecting CK components



Universal program workflow to compile, run, profile and autotune diverse benchmarks across diverse data sets and platforms, validate results, record experiments, share and reproduce them, and report discrepancies

cKnowledge.org/repo

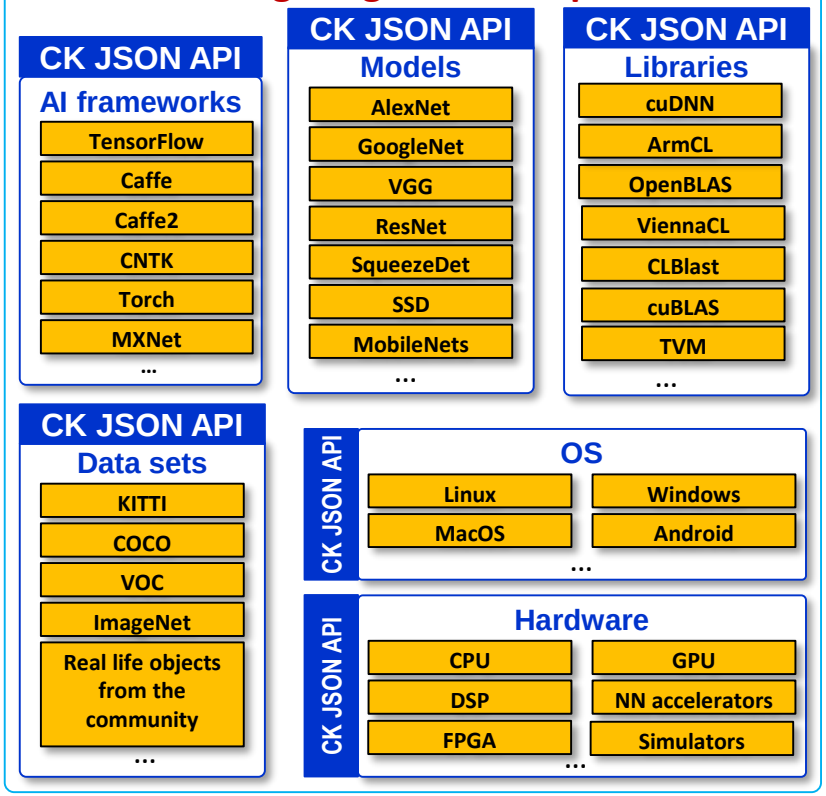
```
$ ck pull repo:ck-crowdtuning

$ 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
```

We can even automatically generate reproducible and interactive articles (collaboration with Raspberry Pi foundation): cKnowledge.org/rpi-crowd-tuning

Enabling open science: cKnowledge.org/partners

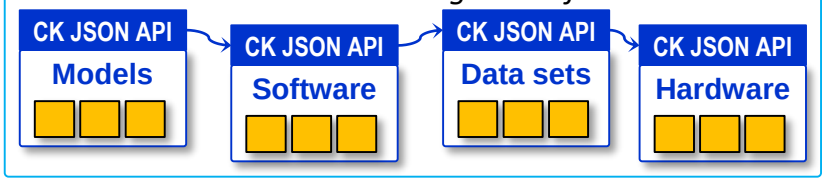
Repositories of customizable, portable and reusable research components with CK API

cKnowledge.org/shared-repos.html



Customizable CK workflows for real-world user tasks

Assemble scenarios such as image classification as LEGO™



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

Crowdsource experiments with the help of volunteers across diverse models, data sets and platforms

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

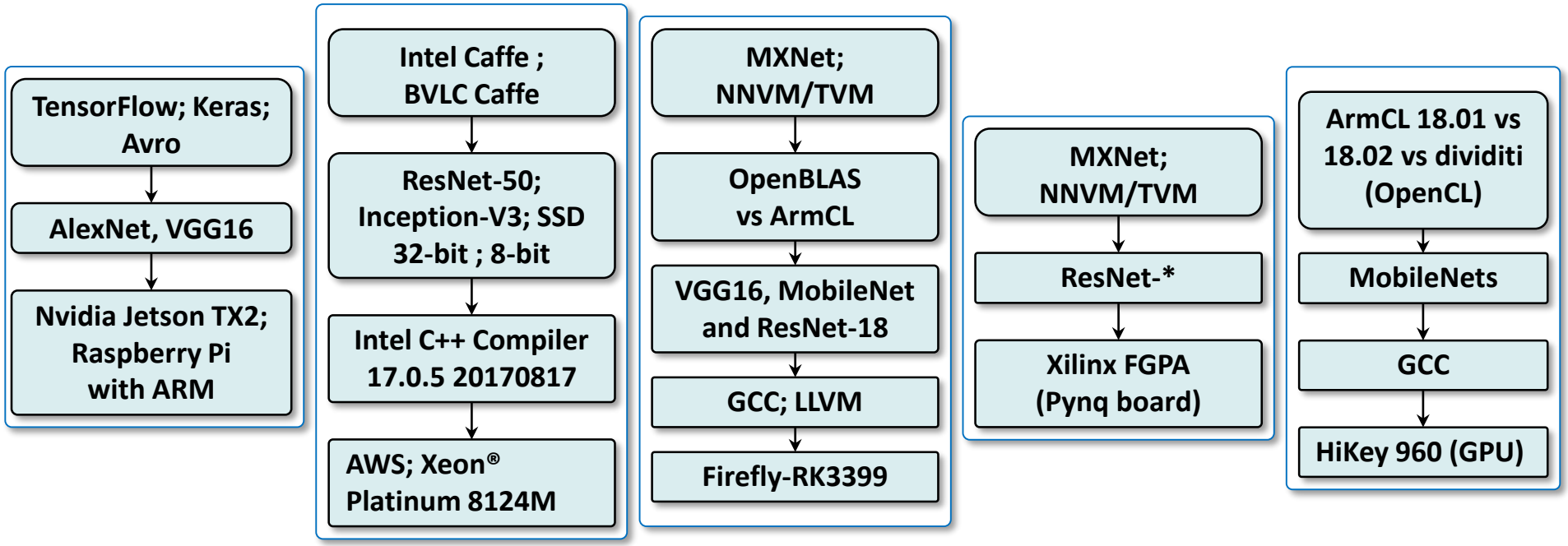
cKnowledge.org/repo

Help students learn multidisciplinary techniques, quickly prototype new ones, validate them in practice with companies, and even contribute back new research components

Help companies select the most appropriate workflows, save R&D costs, accelerate adoption of new techniques!

cKnowledge.org/request: the 1st reproducible tournament at ACM ASPLOS'18

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



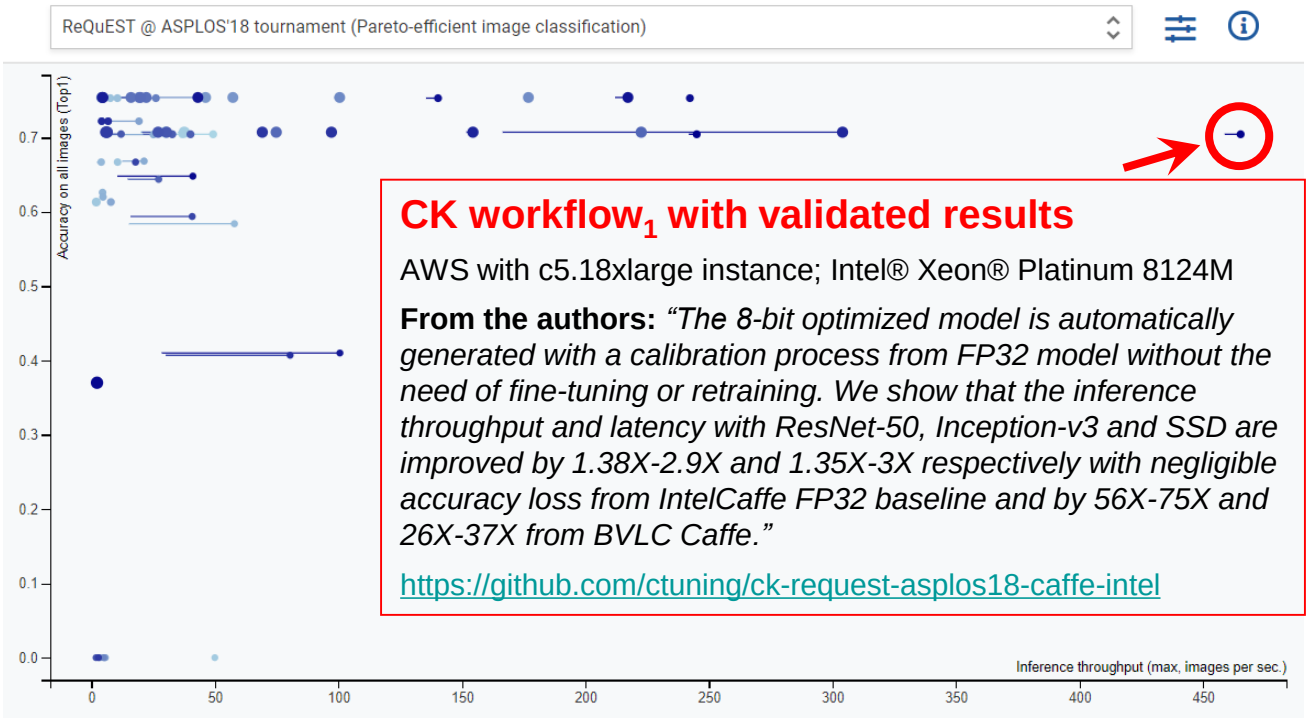
Public validation at github.com/ctuning/ck-request-asplos18-results via GitHub issues.

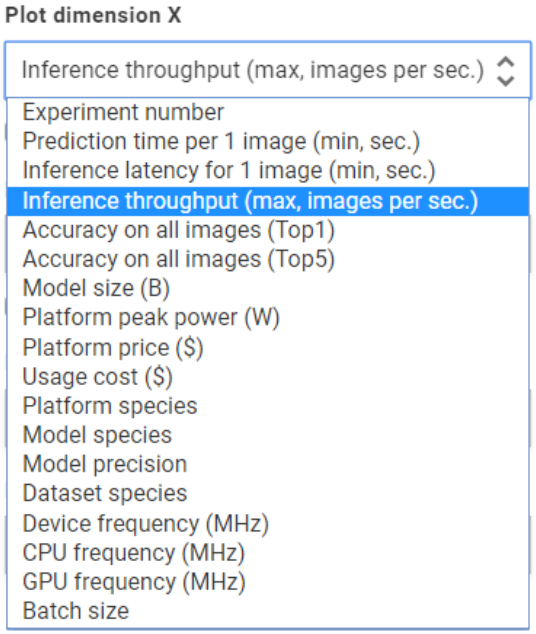
All validated papers are published in the ACM DL
with **portable, customizable and reusable CK components and workflows**:
dl.acm.org/citation.cfm?doid=3229762

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

All results and research components are available via a live CK scoreboard

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!



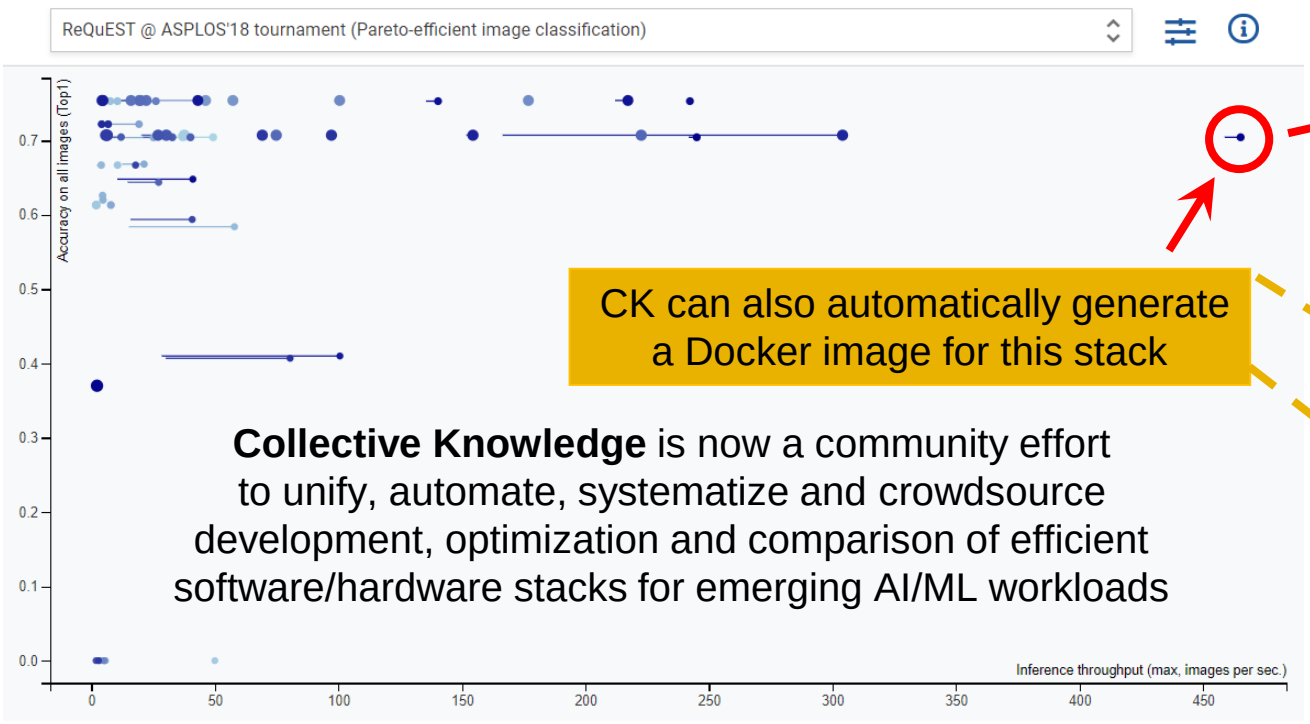


We are not announcing a single winner! We show all multi-dimensional results at cKnowledge.org/dashboard/request.asplos18

and let users select best ML/SW/HW stacks depending on multiple constraints!

Amazon colleagues managed to reproduce results and started using CK

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

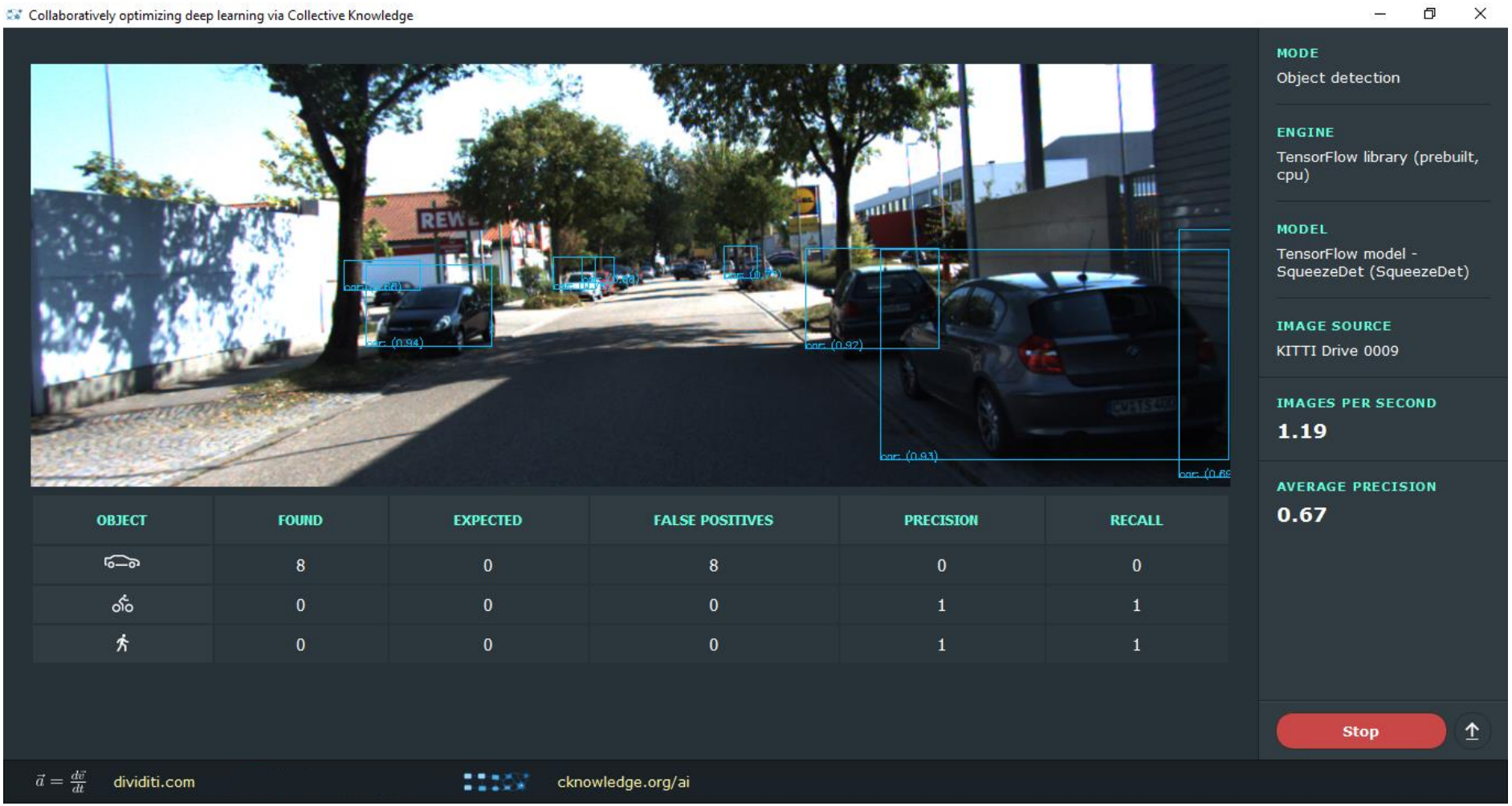
amazon web services

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

See Amazon presentation at O'Reilly AI conference:

conferences.oreilly.com/artificial-intelligence/ai-eu/public/schedule/detail/71549

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



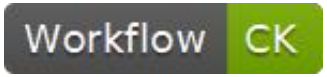
OBJECT	FOUND	EXPECTED	FALSE POSITIVES	PRECISION	RECALL
car	8	0	8	0	0
cyclist	0	0	0	1	1
pedestrian	0	0	0	1	1

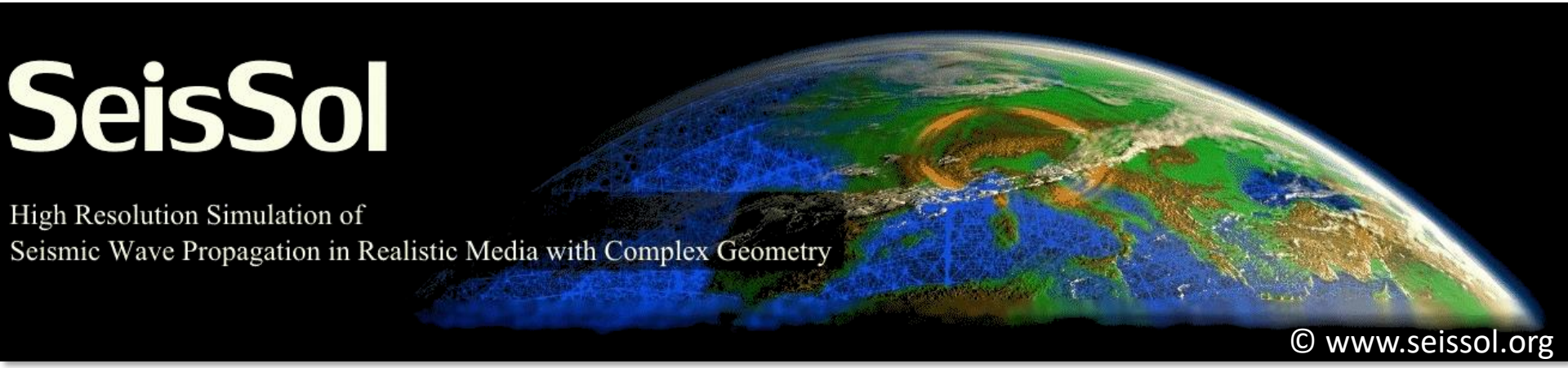
Performance, accuracy, power consumption practically never match official reports!

CK allows to select the most efficient SW/HW stacks on a Pareto frontier (performance, accuracy, energy, memory usage, costs) for object detection, image classification and other tasks: www.youtube.com/watch?v=1ldgVZ64hEI

CK helps to automate Student Cluster competitions

github.com/ctuning/ck-scc18/wiki - proof-of-concept CK workflow
to automate installation, execution and customization of SeisSol application
from the SC18 SCC Reproducibility Challenge
across different platforms, environments and datasets



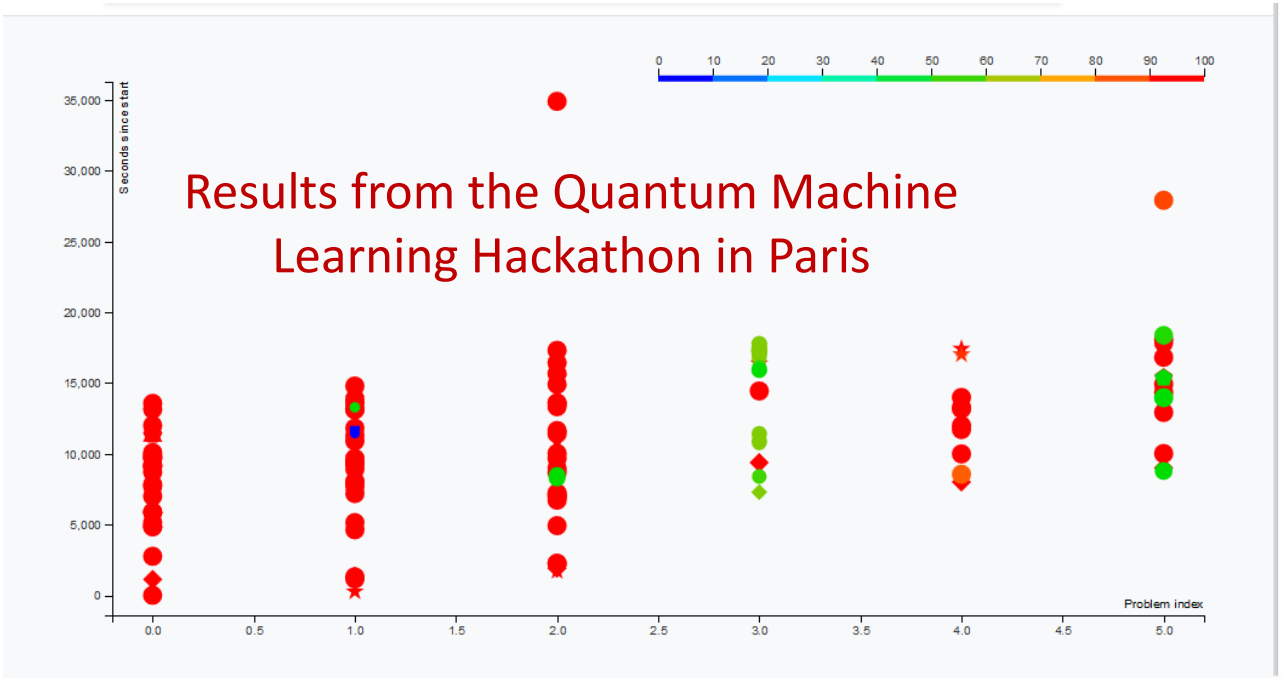


- Support automatic detection of already installed tools and data sets
- Can install missing dependencies via EasyBuild and Spack
- Can deploy application on different supercomputers with different job managers
- Can automatically validate the correctness of results (output, performance)

CK is used to collaboratively advance quantum computing

cKnowledge.org/quantum - Quantum Collective Knowledge workflows (QCK) to support reproducible hackathons, and help researchers share, compare and optimize different algorithms across conventional and quantum platforms

cKnowledge.org/dashboard/hackathon.20190127



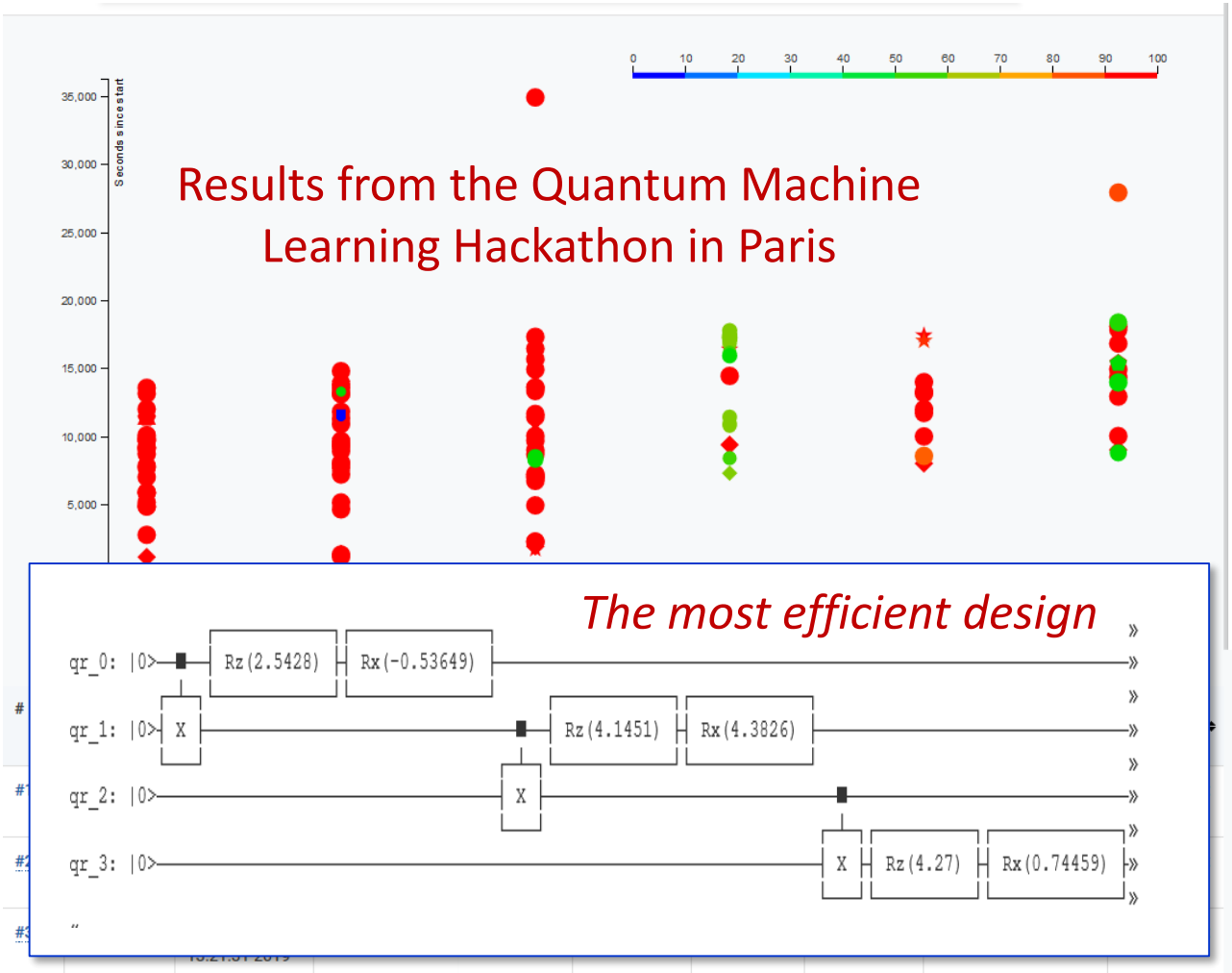
#	Problem index	Timestamp (UTC)	Team name	Training time (sec)	Training accuracy	Test accuracy	Solution's rank	Source code	Quantum circuit
#1	4	Sun Jan 27 12:19:42 2019	Optimize, adapt, overcome	47.20	100.0	100.0	1	continuous_solver	Show circuit
#2	4	Sun Jan 27 12:52:49 2019	prevision.io	80.68	100.0	100.0	2	continuous_solver	Show circuit
#3	4	Sun Jan 27 13:21:31 2019	rebecca	171.54	100.0	100.0	3	continuous_solver	Show circuit

$\frac{d\vec{v}}{dt}$ IBM rigetti RIVERLANE ThoughtWorks® QUANTO NATION Innovate UK

CK is used to collaboratively advance quantum computing

cKnowledge.org/quantum - Quantum Collective Knowledge workflows (QCK) to support reproducible hackathons, and help researchers share, compare and optimize different algorithms across conventional and quantum platforms

cKnowledge.org/dashboard/hackathon.20190127



$\frac{d\vec{v}}{dt}$

IBM

rigetti

RIVERLANE

ThoughtWorks®

QUANTO NATION

Innovate UK

cKnowledge.org: future plans to enable open science

Websites:

- github.com/ctuning/ck
- http://cKnowledge.org
- cKnowledge.org/shared-repos.html

Huge thanks to all partners and contributors:

http://cKnowledge.org/partners

From prototype to production quality (beginning of a long journey)

- Collaboratively standardize APIs and meta descriptions
- Improve installation and documentation; add GUIs
- Add more CK components and workflows for real-world tasks
- Create online index (everyone could describe their components)

Open to collaboration

- Joint R&D projects, hackathons, and tournaments (AI, ML, quantum)
- Automation and sharing of experiments
- Reproducible articles with reusable workflows

Contact: Grigori.Fursin@cTuning.org or grigori@dividiti.com