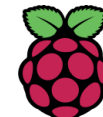
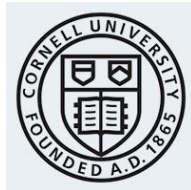


Collective Knowledge (CK): an open-source framework to automate, reproduce, and crowdsource HPC experiments

github.com/ctuning/ck

cknowledge.org/partners

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



Raspberry Pi



UNIVERSITY OF
TORONTO

CK complements many popular tools and services



GitHub



docker



slurm
workload manager



ACM DL DIGITAL
LIBRARY



Azure



figshare



Grigori Fursin

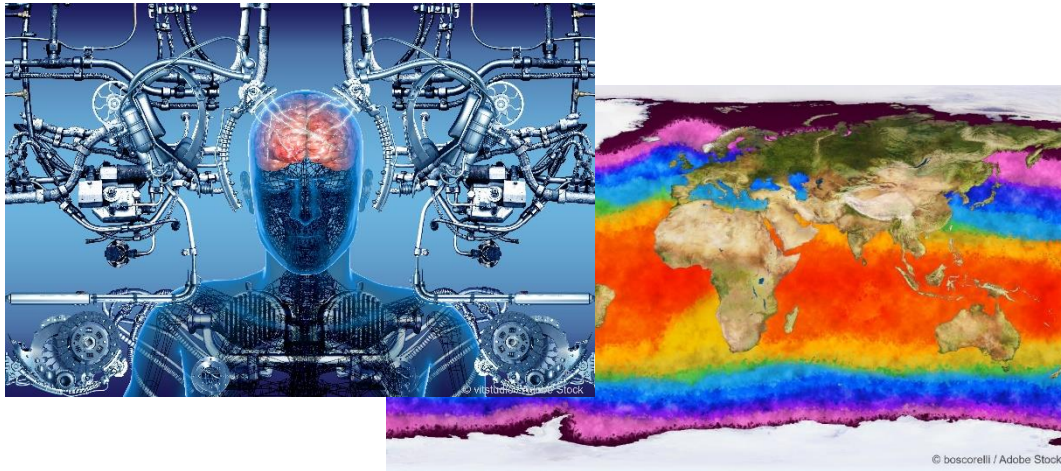
@grigori_fursin

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 services

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

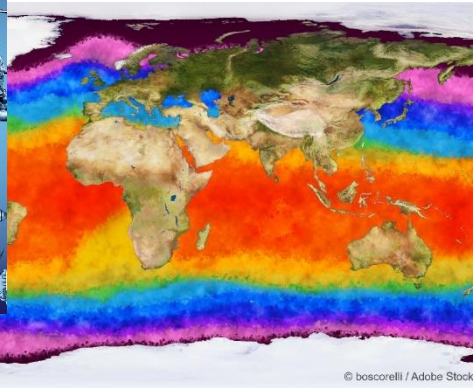
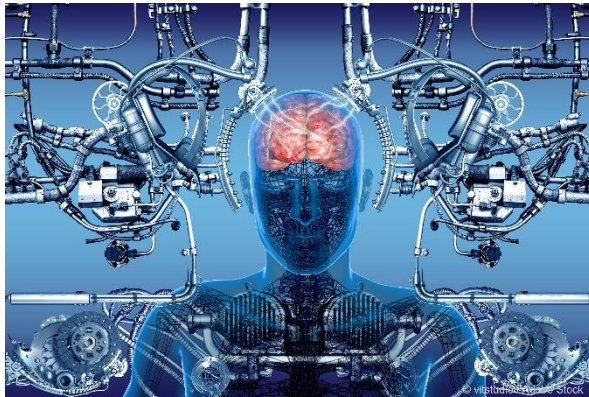
- Linux
- MacOS
- BSD
- Windows
- Android

Shells

- bash
- sh
- csh
- ksh
- Windows shell

Programs

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



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

Hardware

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

Platforms

- HPC
- Desktops
- IoT
- Mobile
- Cloud services

Let's innovate ...

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

Languages

- C++
- C#
- C
- Go

Compilers

- LLVM
- GCC
- Intel
- PGI

DevOps tools

- Git
- Jenkins
- Docker

Package managers

- Anaconda
- Go
- Nnm

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

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 services



© Elnur / Adobe Stock

managers

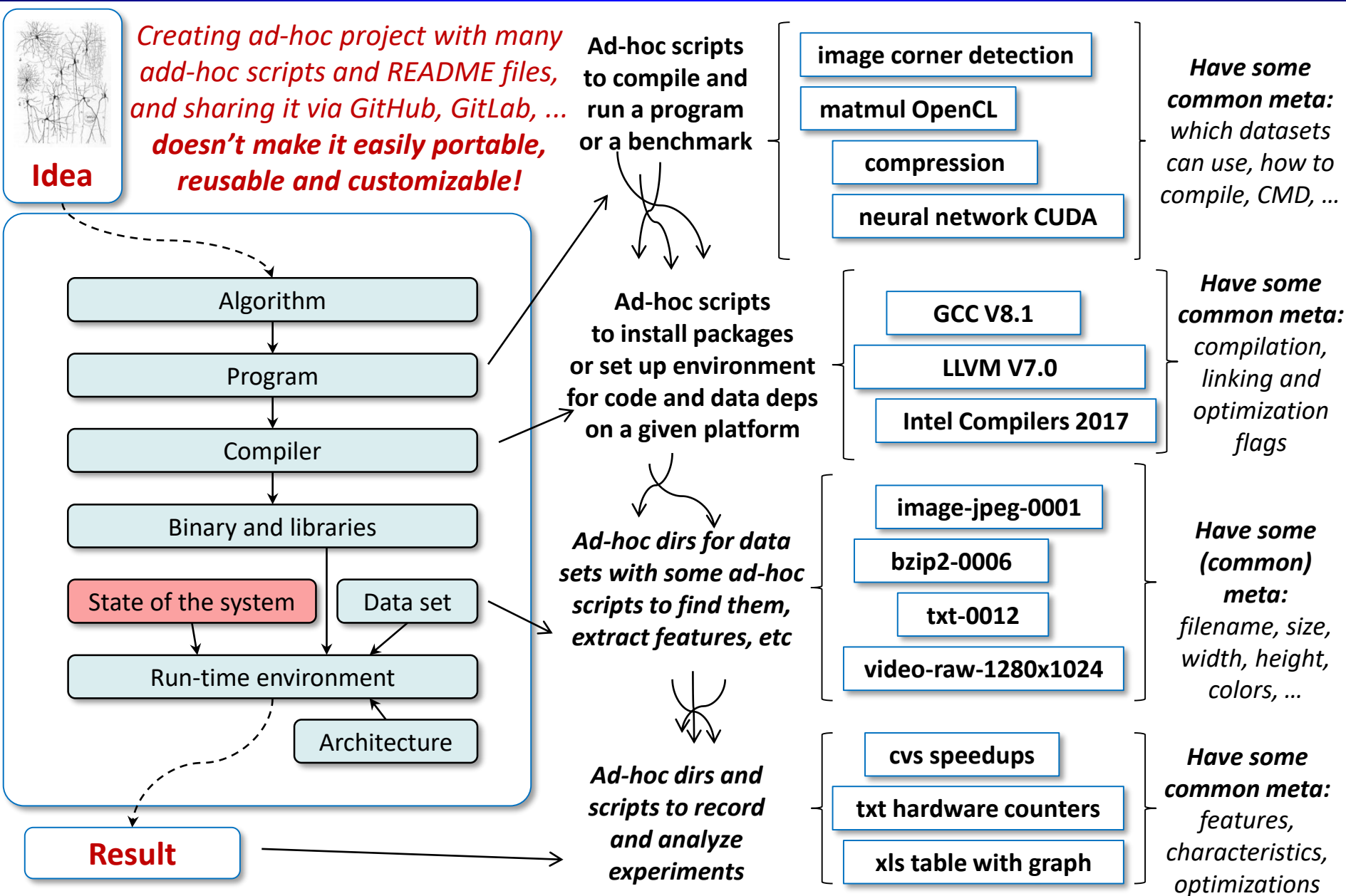
- MiDataSets
- Human Cell Atlas
- 1000 Genomes
- Earth models
- OpenStreetMap

- MPI
- SLURM
- PBS
- FLUX

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

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

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



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, etc.

Ad-hoc scripts to compile and

image corner detection

OpenCL

Have some common meta: which datasets can use, how to write 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!

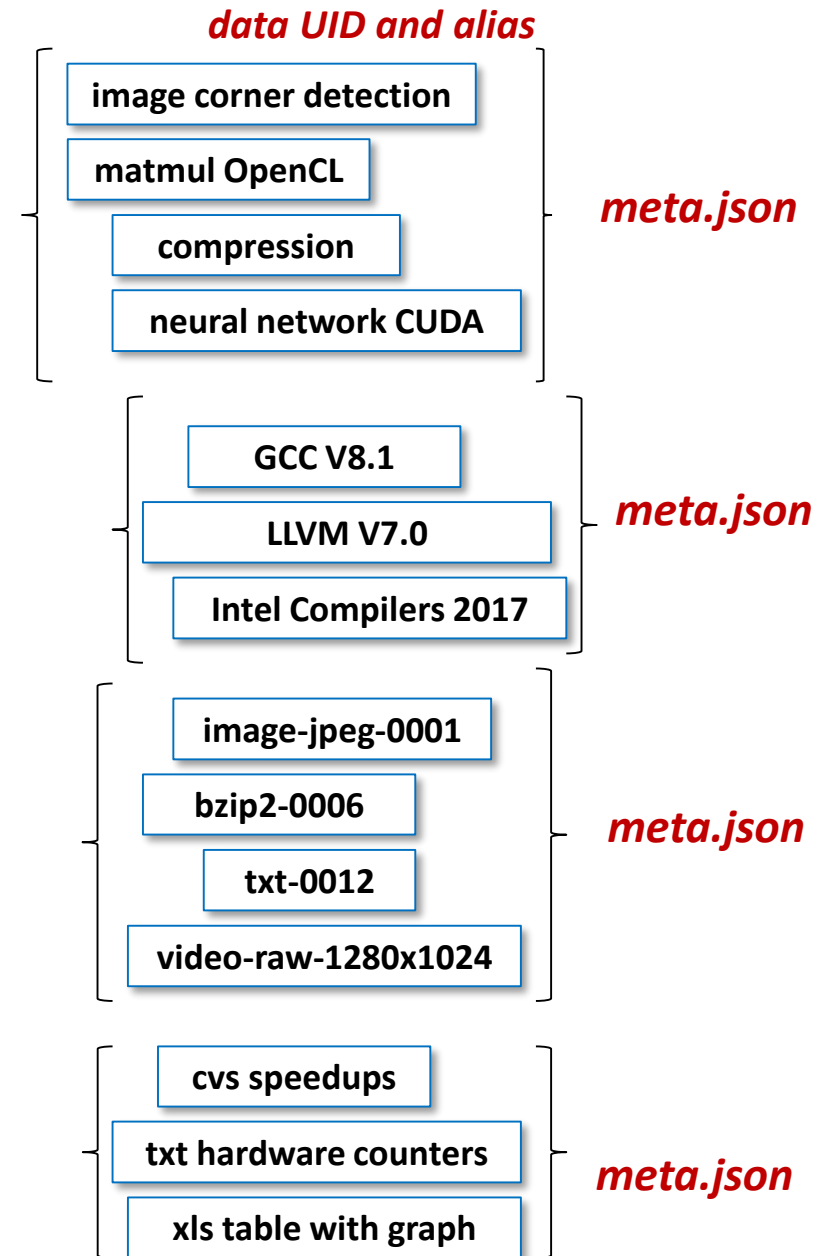
Provide unified Python APIs and JSON meta descriptions for similar code and data objects – enables DevOps!

Python module
“program”
with functions:
compile and run

Python module
“soft”
with function:
detect

Python module
“dataset”
with function:
extract_features

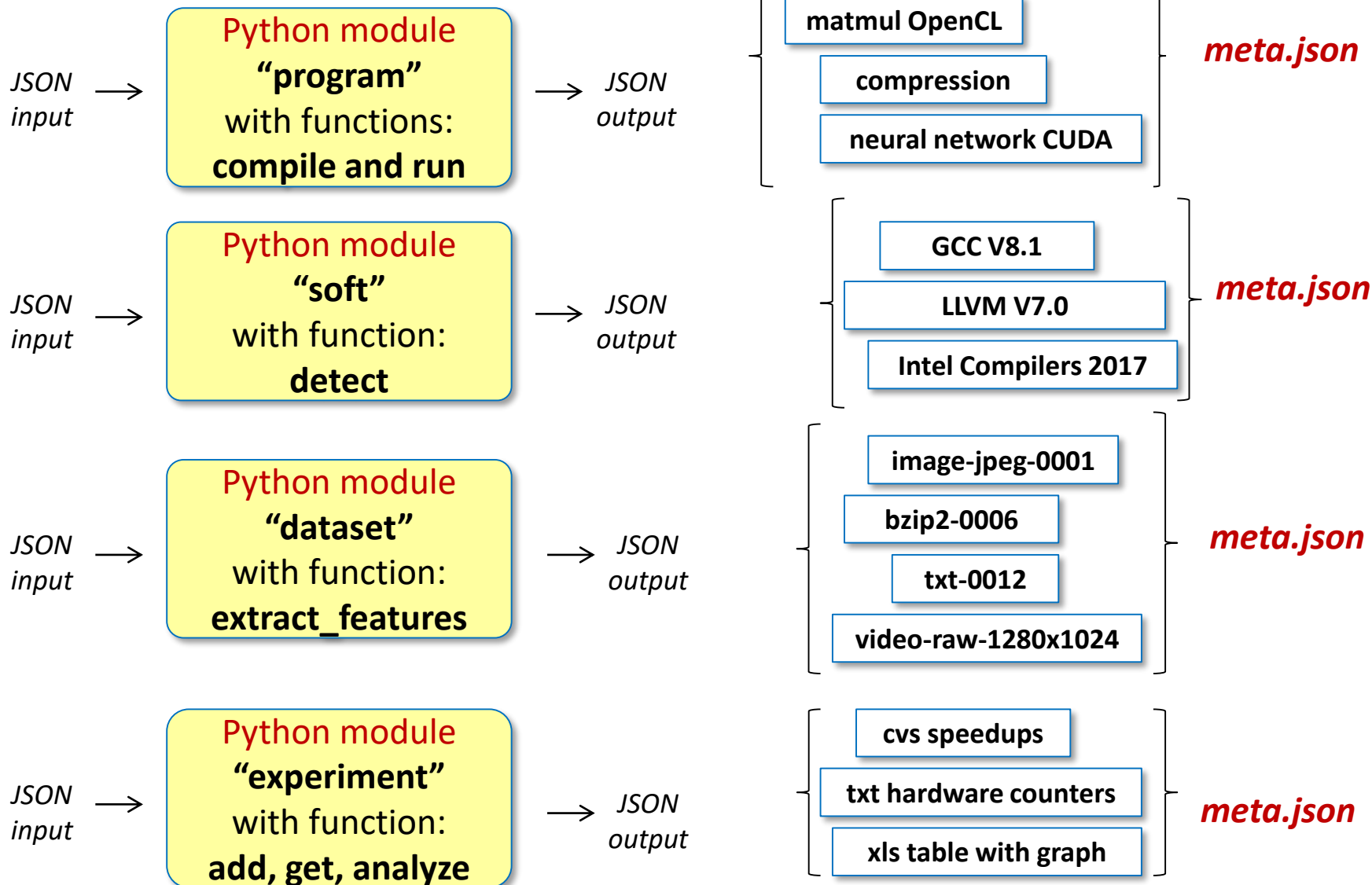
Python module
“experiment”
with function:
add, get, analyze



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

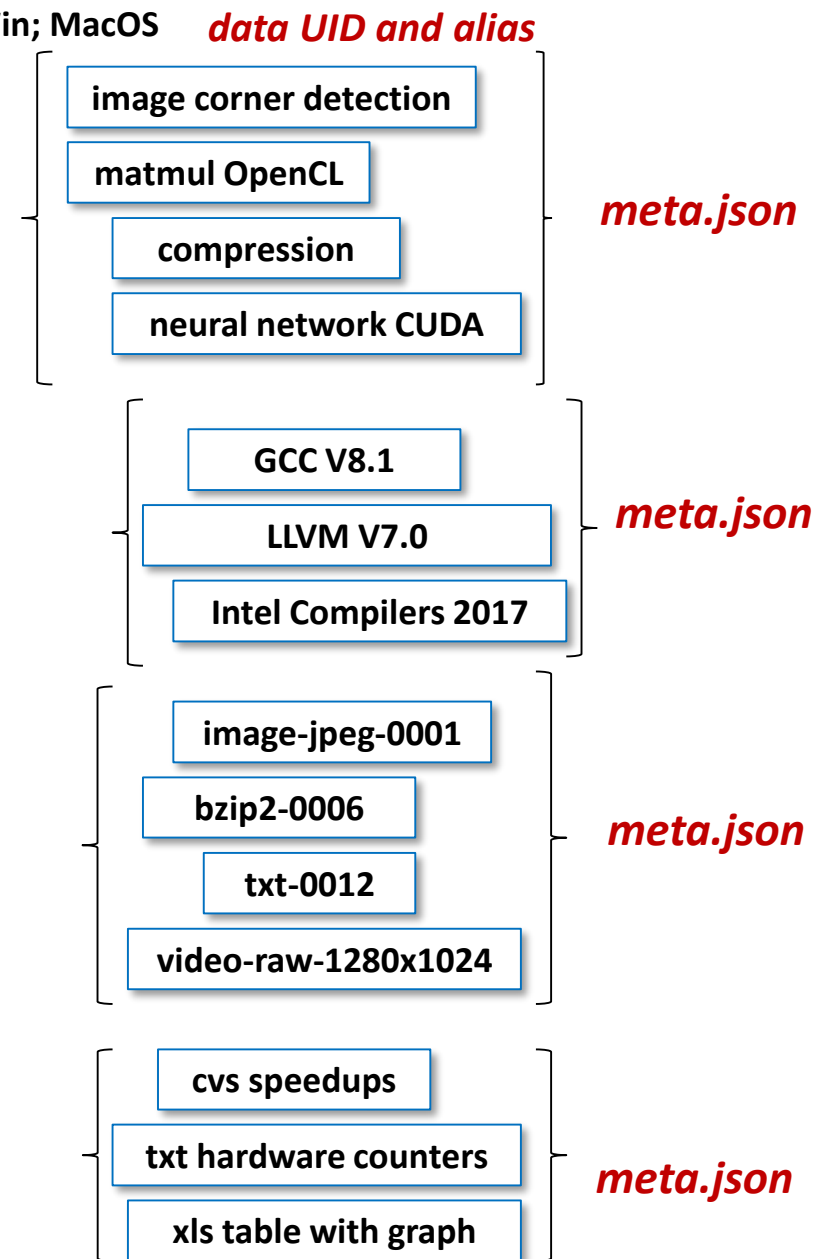
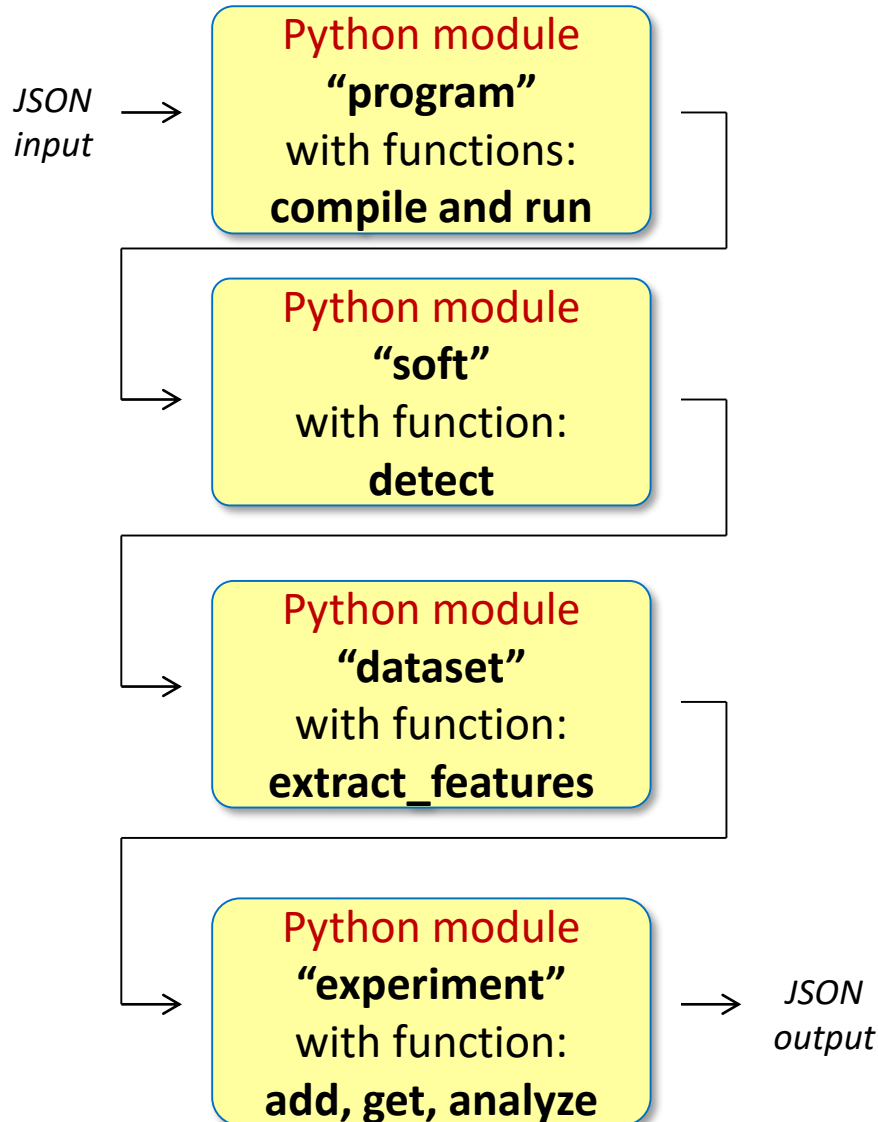
\$ ck {function} {module name}:{data name} @input.json



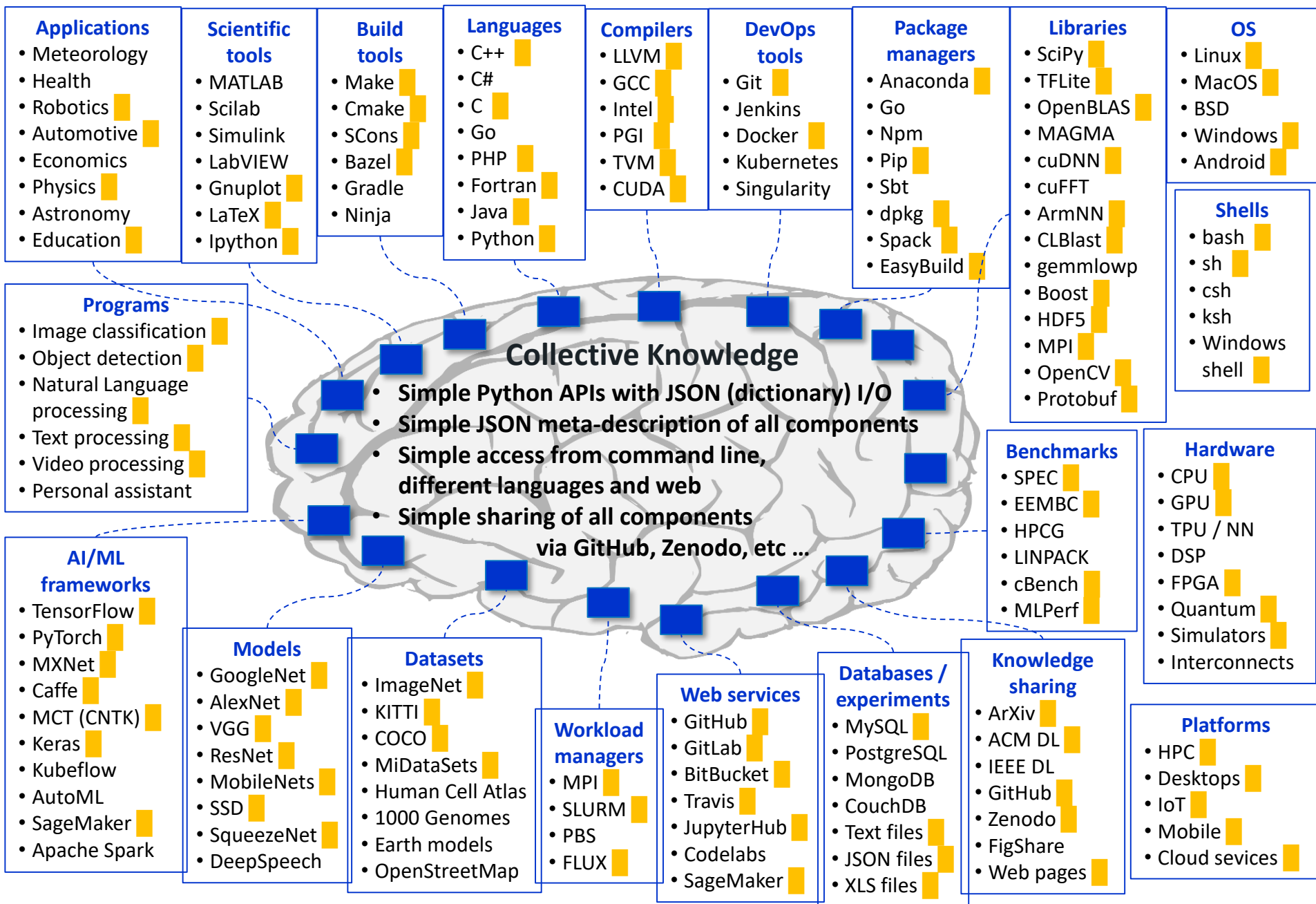
CK framework: the community can implement complex workflows via CK APIs

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

Assemble workflows from shared components



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



cKnowledge.org/shared-repos.html

cKnowledge.org/shared-modules.html

Artifact automated and reusable

Collective Knowledge COMPATIBLE

Workflow CK

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
```

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

```
$ ck detect soft:compiler.llvm
```

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

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

```
$ ck search package --tags=model cKnowledge.org/shared-packages.html
```

```
$ ck install compiler:compiler-llvm-7.0.0-universal
```

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

```
$ ck virtual env --tags=llvm,v7.0.0
```

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

Enabling customizable and portable workflows by connecting CK components

Common JSON API

JSON
meta

Algorithm / source code

AI framework

Available libraries / skeletons

Compilers

Binary or byte code

Inputs

Various models

Run-time environment

Run-time state
of the system

Hardware,
simulators

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

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
MacOS
Windows
Android
...

CK JSON API

Hardware

CPU
DSP
FPGA
GPU
NN accelerators
Simulators
...

Customizable CK workflows
for real-world user 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 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

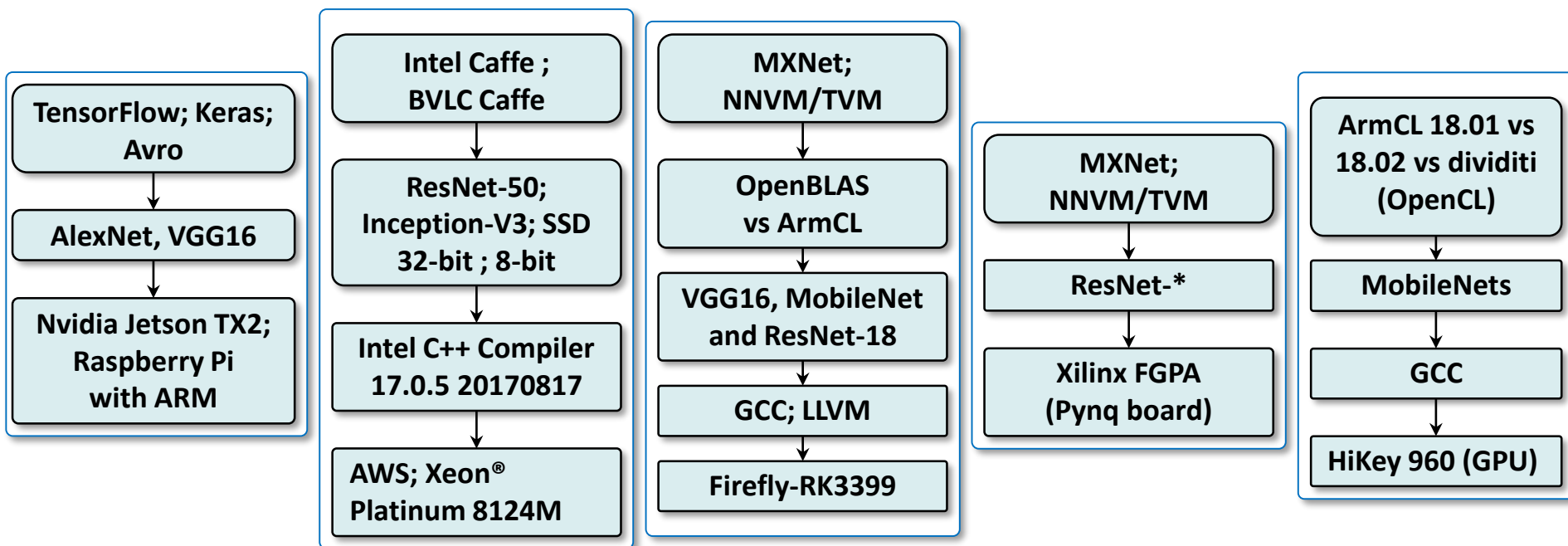
cKnowledge.org/repo

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.

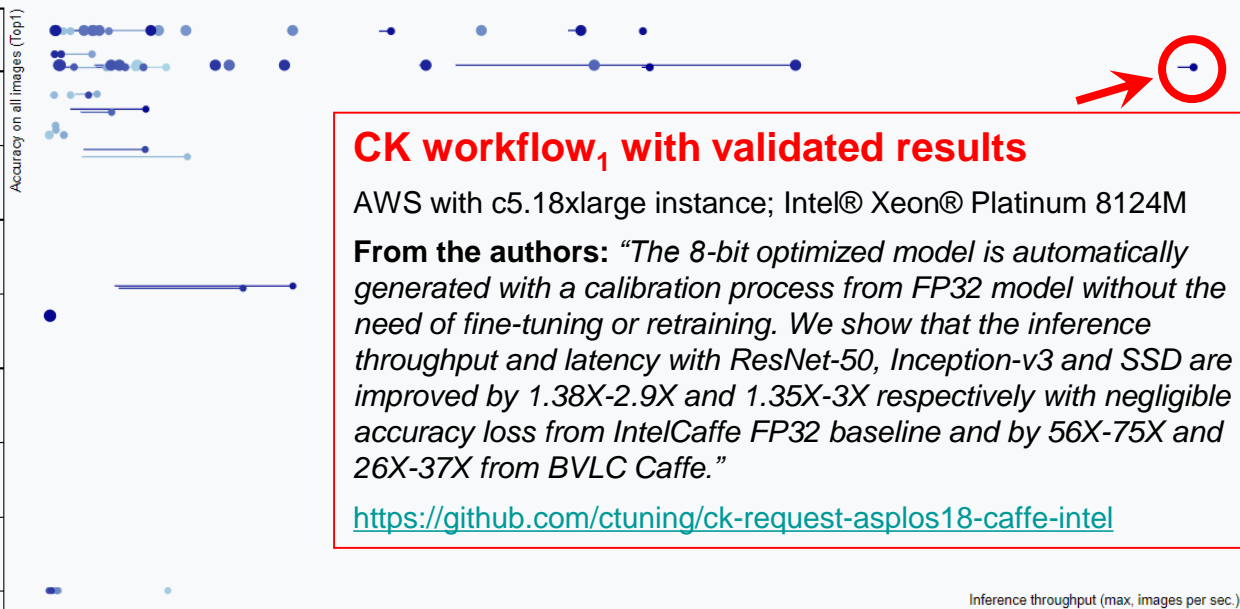
All validated papers are published in the ACM DL
with **portable, customizable and reusable CK components and workflows**:
dl.acm.org/citation.cfm?doid=3229762

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!

ReQuEST @ ASPLOS'18 tournament (Pareto-efficient image classification)



CK workflow₁ with validated results

AWS with c5.18xlarge instance; Intel® Xeon® Platinum 8124M

From the authors: *"The 8-bit optimized model is automatically generated with a calibration process from FP32 model without the need of fine-tuning or retraining. We show that the inference throughput and latency with ResNet-50, Inception-v3 and SSD are improved by 1.38X-2.9X and 1.35X-3X respectively with negligible accuracy loss from IntelCaffe FP32 baseline and by 56X-75X and 26X-37X from BVLC Caffe."*

<https://github.com/ctuning/ck-request-asplos18-caffe-intel>

Plot dimension X

Inference throughput (max, images per sec.)

Experiment number

Prediction time per 1 image (min, sec.)

Inference latency for 1 image (min, sec.)

Inference throughput (max, images per sec.)

Accuracy on all images (Top1)

Accuracy on all images (Top5)

Model size (B)

Platform peak power (W)

Platform price (\$)

Usage cost (\$)

Platform species

Model species

Model precision

Dataset species

Device frequency (MHz)

CPU frequency (MHz)

GPU frequency (MHz)

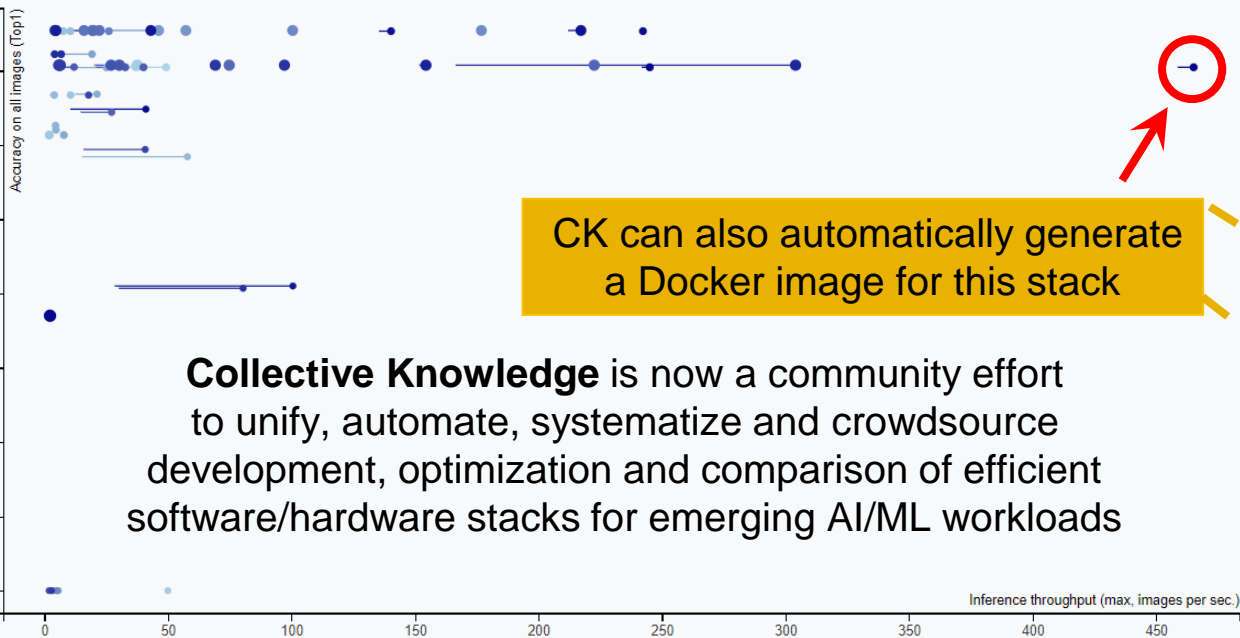
Batch size

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!

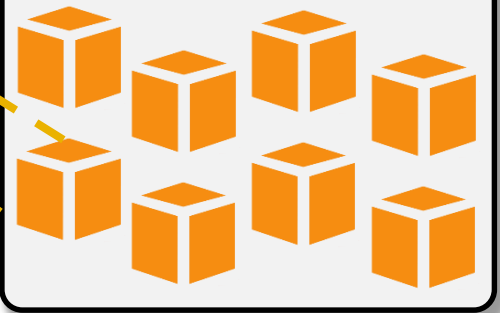
Amazon colleagues managed to reproduce results and started using CK

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!

ReQuEST @ ASPLOS'18 tournament (Pareto-efficient image classification)



CK assists
AWS market place
with collaboratively
optimized AI/ML stacks



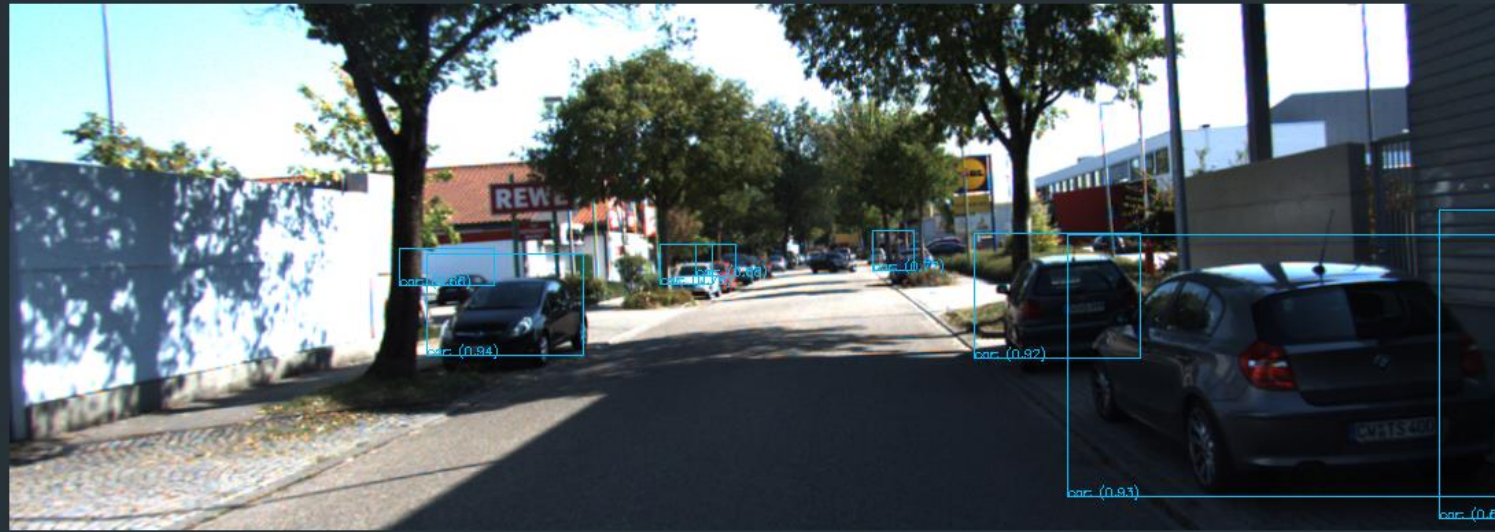
Accelerate technology transfer: companies can now quickly validate published techniques in their production environment using shared CK workflows!




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

Collaboratively optimizing deep learning via Collective Knowledge



OBJECT	FOUND	EXPECTED	FALSE POSITIVES	PRECISION	RECALL
	8	0	8	0	0
	0	0	0	1	1
	0	0	0	1	1

MODE

Object detection

ENGINE

TensorFlow library (prebuilt, cpu)

MODEL

TensorFlow model - SqueezeDet (SqueezeDet)

IMAGE SOURCE

KITTI Drive 0009

IMAGES PER SECOND

1.19

AVERAGE PRECISION

0.67

Stop



$\vec{a} = \frac{d\vec{v}}{dt}$ dividiti.com



cknowledge.org/ai

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

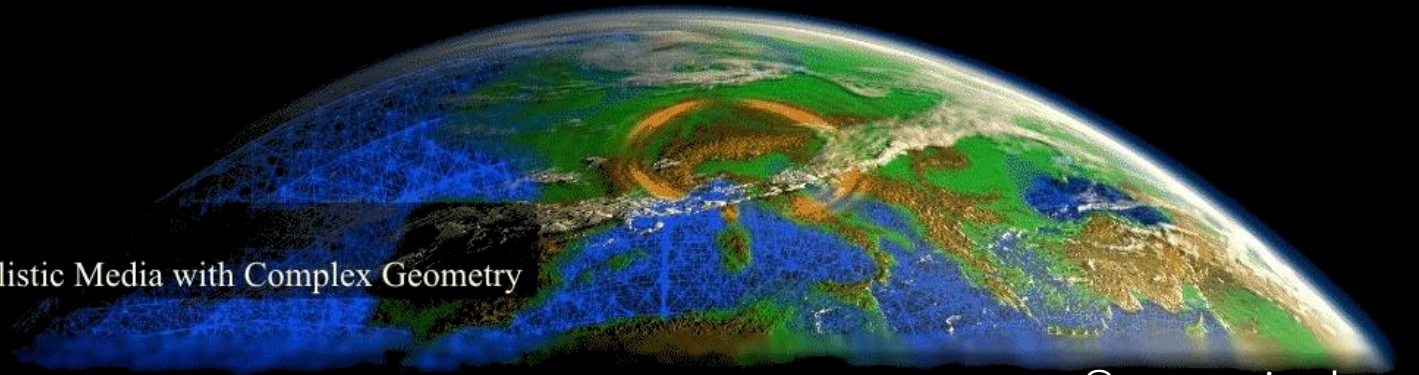
Artifact automated and reusable

Collective Knowledge COMPATIBLE

Workflow CK

SeisSol

High Resolution Simulation of
Seismic Wave Propagation in Realistic Media with Complex Geometry



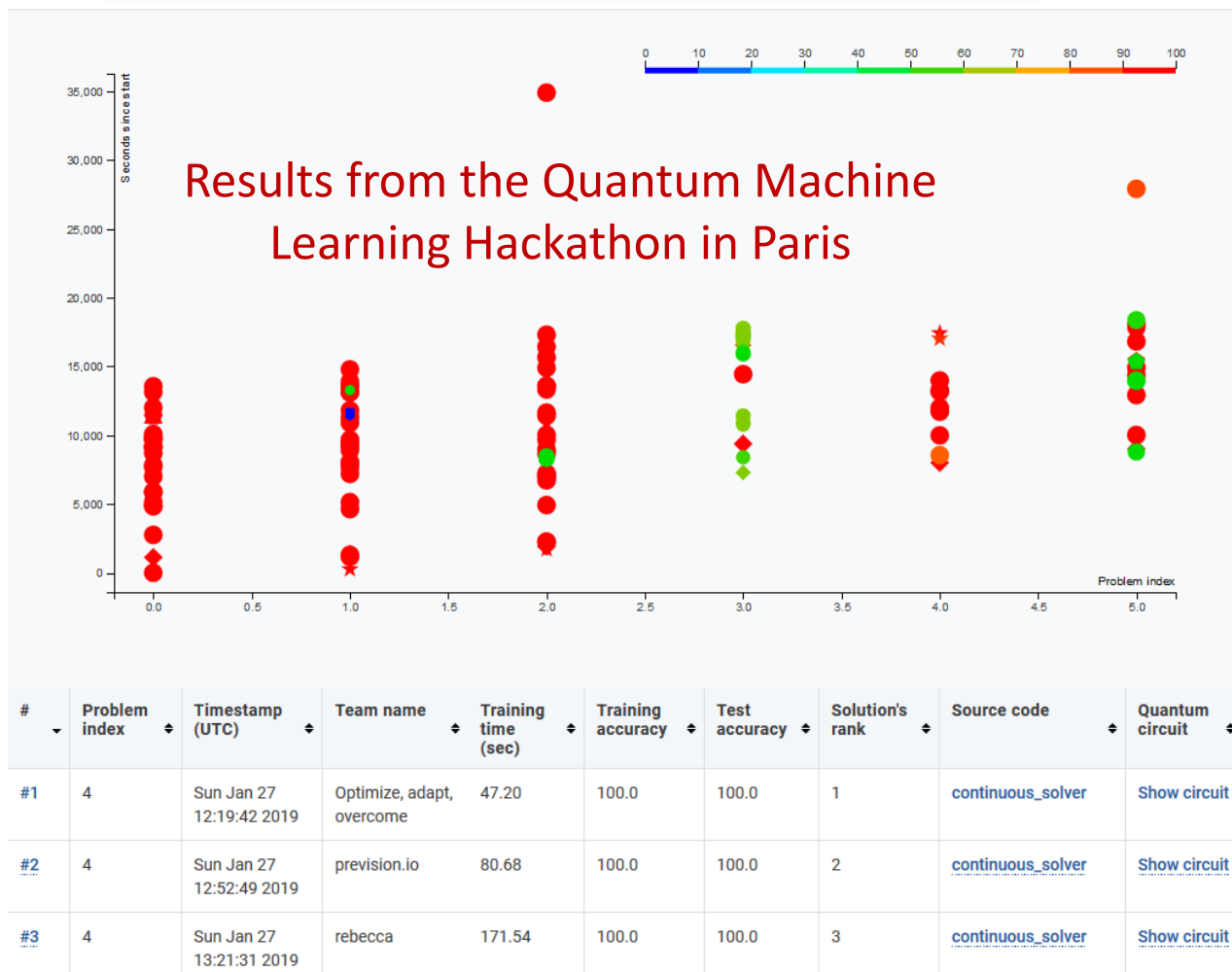
© www.seissol.org

- 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



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

rigetti

RIVERLANE

ThoughtWorks®

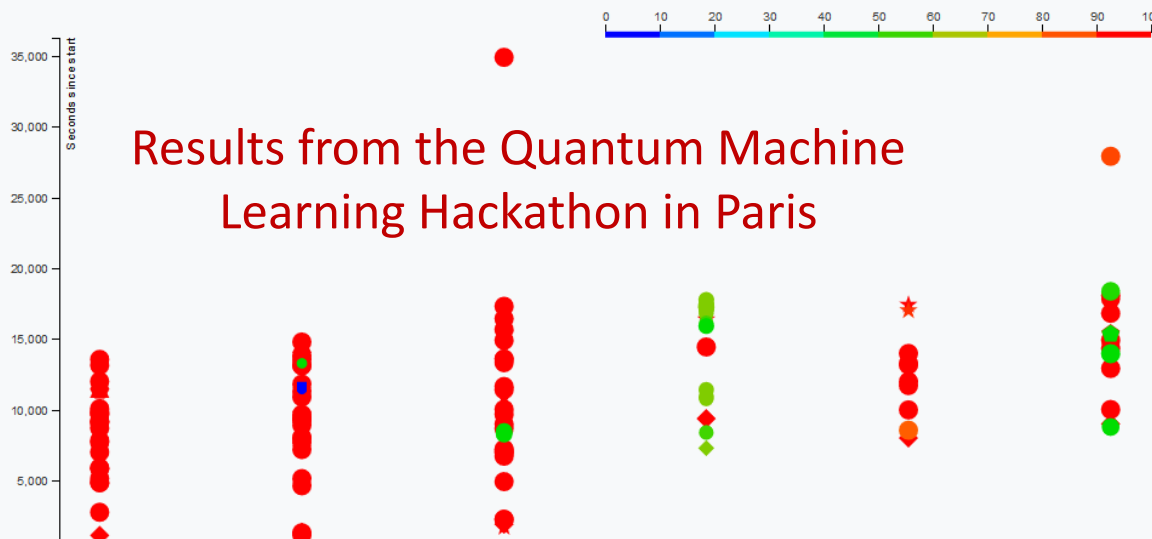
QUANTO
NATION

Innovate UK

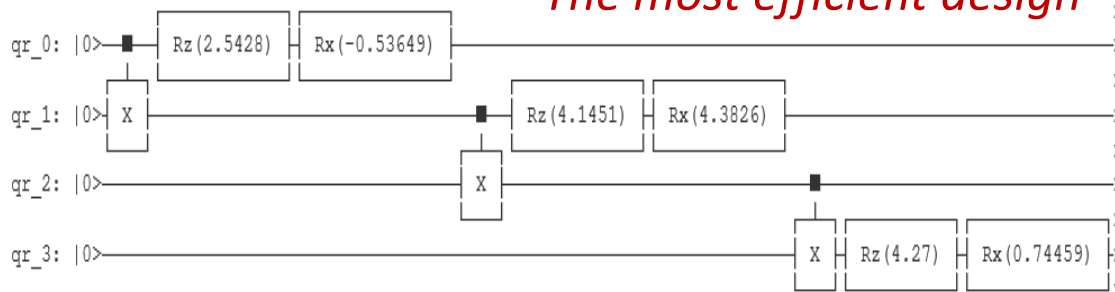
CK is used to collaboratively advance quantum computing

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



The most efficient design



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

rigetti

RIVERLANE

ThoughtWorks®

QUANTO
NATION

Innovate UK

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