Towards Reproducible and Reusable Deep Learning Systems Research Artifacts

Thierry Moreau[1], **Anton Lokhmotov**[2], **Grigori Fursin**[2,3]
[1]University of Washington, USA, [2]dividiti, UK, [3]cTuning foundation, France

Abstract

This paper discusses results and insights from the 1st ReQuEST workshop, a collective effort to promote reusability, portability and reproducibility of deep learning research artifacts within the Architecture/PL/Systems communities. ReQuEST (Reproducible Quality-Efficient Systems Tournament) exploits the open-source Collective Knowledge framework (CK) to unify benchmarking, optimization, and co-design of deep learning systems implementations and exchange results via a live multi-objective scoreboard. Systems evaluated under ReQuEST are diverse and include an FPGA-based accelerator, optimized deep learning libraries for x86 and ARM systems, and distributed inference in Amazon Cloud and over a cluster of Raspberry Pis. We finally discuss limitations to our approach, and how we plan improve upon those limitations for the upcoming SysML artifact evaluation effort.

1 ReQuEST Overview

The quest to continually optimize deep learning systems has introduced new deep learning models, frameworks, DSLs, libraries, compilers and hardware architectures. In this frantically changing environment, is has become critical to quickly reproduce, deploy, and build on top of existing research. While open-sourcing research artifacts is one step in the right direction, it is not sufficient to guarantee ease of *reproducibility* and *reusability*. To enable reproducible and reusable research, we need to provide complete, customizable, and portable *workflows* that combine off-the-shelf and custom layers of the system stack and deploys them in a push-button fashion to generate end-to-end metrics of importance.

In an effort to promote reproducible, reusable, and portable workflows in deep learning systems research, we introduced the ReQuEST workshop at the ACM ASPLOS 2018 (for multidisciplinary systems research spanning computer architecture and hardware, programming languages and compilers, operating systems and networking). The goal was to have computer architects, compilers, and systems researchers submit deep learning research artifacts (code, data, and experiments) using a unified Collective Knowledge (CK) workflow framework Fursin et al. (2016) to produce a *multi-objective scoreboard* that would rank submissions under varied cost metrics that include: ImageNet validation (50,000 images), latency (seconds per image), throughput (images per second), platform price (dollars), and peak power consumption (Watts). To keep the task of collecting artifacts tractable, we focused on a single problem: ImageNet classification, but gave complete freedom over what models, frameworks, libraries, compilers and hardware platforms were being used to solve the classification problem.

The most important difference of ReQuEST from other related workshops and tournaments such as DawnBench daw (2018) and LPIRC lpi (2015) is that we not only publish final results but also share portable and customizable workflows (i.e. not just Docker images) with all related research components (models, data sets, libraries) to let the community immediately reuse, improve, and build upon them.

Submitted to Machine Learning Open Source Software (MLOSS, co-located at NIPS 2018). Do not distribute.

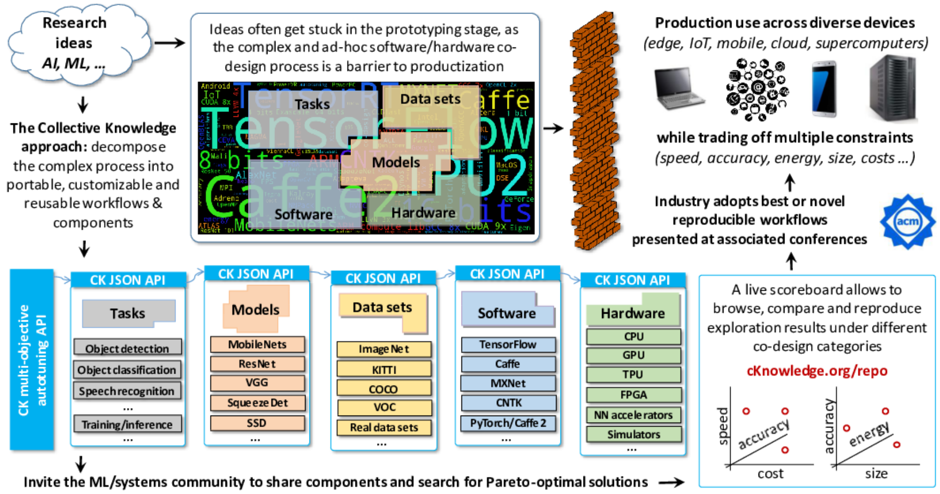

Figure 1: We leverage the open Collective Knowledge workflow framework (CK) and the rigorous ACM artifact evaluation methodology (AE) to allow the community collaboratively explore quality vs. efficiency trade-offs for rapidly evolving workloads across diverse systems.

The first iteration of the ReQuEST workshop led to five artifact submissions that were unified under the CK framework and evaluated (reproduced) by the organizers. What the submissions lacked in quantity, they made up for in terms of diversity: (1) submissions spanned architecture, compilers, and systems research, (2) utilized x86, ARM, and FPGA-based platforms; and (3) were deployed on single-node systems as well as distributed nodes.

2 Unifying Artifacts and Workflows with CK

ReQuEST aims to promote reproducibility of experimental results and reusability/customization of systems research artifacts by standardizing evaluation methodologies and facilitating the deployment of efficient solutions on heterogeneous platforms. For that reason, packaging artifacts (scripts, libraries, frameworks, data sets, models) and experimental results requires a bit more involvement than sharing some CSV/JSON files or checking out a given GitHub repository. That is why we build our competition on top of CK Fursin et al. (2016) to provide unified evaluation and a real-time leader-board of submissions. CK is an open-source portable workflow framework, used as standard ACM artifact evaluation methodology from ACM and IEEE systems conferences (CGO, PPoPP, PACT, SuperComputing).

CK works a Python wrapper framework to help users share their code and data as customizable and reusable plugins with a common JSON API, meta description and an integrated package manager, adaptable to a user platform with Linux, Windows, MacOS and Android. Researchers can then quickly prototype experimental workflows from shared components, crowdsource benchmarking and autotuning across diverse models, data sets and platforms, exchange results via public scoreboards, and generate interactive reports ck- (2018).

3 Artifact Submissions Overview

The ReQuEST-ASPLOS'18 proceedings, available in the ACM Digital Library, include five papers with Artifact Appendices and a set of ACM reproducibility badges.

The CK repository for all ReQuEST-ASPLOS'18 artifacts are documented and available at the following link: https://github.com/ctuning/ck-request-asplos18-results. The interactive live scoreboard can be accessed under the followig URL: http://cKnowledge.org/request-results. The proceedings are accompanied by snapshots of Collective Knowledge workflows covering a very diverse model/software/hardware stack:

- **Models:** MobileNets, ResNet-18, ResNet-50, Inception-v3, VGG16, AlexNet, SSD.

- **Data types:** 8-bit integer, 16-bit floating-point (half), 32-bit floating-point (float).
- **AI frameworks and libraries:** MXNet, TensorFlow, Caffe, Keras, Arm Compute Library, cuDNN, TVM, NNVM.
- **Platforms:** Xilinx Pynq-Z1 FPGA, Arm Cortex CPUs and Arm Mali GPGPUs (Linaro HiKey960 and T-Firefly RK3399), a farm of Raspberry Pi devices, NVIDIA Jetson TX1 and TX2, and Intel Xeon servers in Amazon Web Services, Google Cloud and Microsoft Azure.

The community can now access all the above CK workflows under permissive licenses and continue collaborating on them via dedicated ReQuEST'18 GitHub projects. First, the workflows can be automatically adapted to new platforms and environments by either detecting already installed dependencies (e.g. libraries) or rebuilding dependencies via an integrated package manager supporting Linux, Windows, MacOS and Android. Second, the workflows can be customized by swapping in new models, data sets, frameworks, libraries, and so on. Third, the workflows can be extended to expose new design and optimization choices (e.g. quantization), as well as evaluation metrics (e.g. power or memory consumption). Finally, the workflows can be used for collaborative autotuning ("crowd-tuning") to explore huge optimization spaces using devices such as Android phones and tablets, with best solutions being made available to the community on the online CK scoreboard.

4 Lessons Learned and Future Work

Our overwhelmingly positive experience has also allowed us to critically assess limitations to the scalability to our approach. Fair competitive benchmarking between different platforms, frameworks, and models is hard work. It requires carefully considering model equivalence (e.g. performing the same mix of operations), input equivalence (e.g. preprocessing the inputs in the same way), output equivalence (e.g. validating the outputs for each input, not just calculating the usual aggregate accuracy score), etc. Formalizing the benchmarking requirements and encapsulating them in shared CK components (e.g. using a framework-independent model representation such as ONNX) and workflows (e.g. for input conversion and output validation), should help standardize and automate the benchmarking process.

Thorough artifact evaluation can take several person-weeks. Each submitted workflow needs to be studied in detail in its original form and then converted into a common format. However, the more reusable CK components (such as workflows, modules/plugins, packages) are shared by the community, the easier the conversion becomes. For example, we have successfully reused several previously shared components for models, frameworks and libraries, as well as the universal CK workflow for program benchmarking and autotuning. We propose to introduce a new ACM reproducibility badge for such unified "plug&play" components. This could eventually lead to creating a "marketplace" for Pareto-efficient implementations (code and data) shared as portable, customizable and reusable CK components.

Finally, full experimental evaluation can take many days/weeks. The AE committee can collaborate with the authors to determine a *minimally useful scope* for evaluation which would still provide insights to the community. The community can eventually crowdsource full evaluation. In other words, AE can be "staged" with a quick check that the artifacts are "functional" before the camera-ready deadline followed by full evaluation using the ReQuEST methodology. In fact, ReQuEST can grow into a non-profit service to conferences and journals. Sponsorship should help attract experienced full-time evaluators, as well as part-time volunteers, to work on unifying and evaluating artifacts and workflows.

Future Work Our experience at ReQuEST-ASPLOS'18 will be repurposed to organize SysML's AE, but at a larger scale. Our long-term vision is to dramatically reduce the complexity and costs of the development and deployment of AI, ML, and other emerging workloads. We believe that having an open repository (marketplace) of customizable workflows with reusable components helps to bring together the multidisciplinary community to collaboratively co-design, optimize, and autotune computer systems across the full model/software/hardware stack. Systems integrators will also benefit from being able to assemble complete solutions by adapting such reusable components to their specific usage scenarios, requirements, and constraints. We envision that our community-driven approach and decentralized marketplace will help accelerate adoption and technology transfer of novel AI/ML techniques similar to the open-source movement.

References

LPIRC: low-power image recognition challenge. https://rebootingcomputing.ieee.org/lpirc, 2015.

Industrial and academic use-cases of Collective Knowledge. http://cKnowledge.org/partners.html, 2018.

An end-to-end deep learning benchmark and competition image classification. https://dawn.cs.stanford.edu/benchmark/, 2018.

Fursin, G., Lokhmotov, A., and Plowman, E. Collective Knowledge: towards R&D sustainability. In *Proceedings of the Conference on Design, Automation and Test in Europe (DATE'16)*, March 2016. URL https://www.researchgate.net/publication/304010295_Collective_Knowledge_Towards_RD_Sustainability.