CarML: Cognitive ARtifacts for Machine Learning

Abdul Dakkak*, Cheng Li*, Carl Pearson*, Jinjun Xiong†, Wen-Mei Hwu*
{dakkak,cli99,pearson,w-hwu}@illinois.edu, jinjun@us.ibm.com
*University of Illinois Urbana-Champaign, †IBM Research Yorktown

ILLINOIS
C3SR center for cognitive computing systems research

Motivation

CarML (Cognitive ARtifacts for Machine Learning) is an open source distributed platform to easily deploy and benchmark machine learning (ML) and deep learning (DL) frameworks and models across hardware infrastructures, within a common interface. CarML allows ML/DL developers to publish and evaluate their models, users to experiment with published models, and system architects to profile end-to-end workflows to inform system designs.

- For ML/DL users: CarML is a platform allowing users to evaluate and consume ML models and algorithms
- For ML/DL developers: CarML is a deployment platform allowing the public to try their models and gather feedback
- For System Architects: CarML is a benchmarking platform to profile and understand system bottleneck

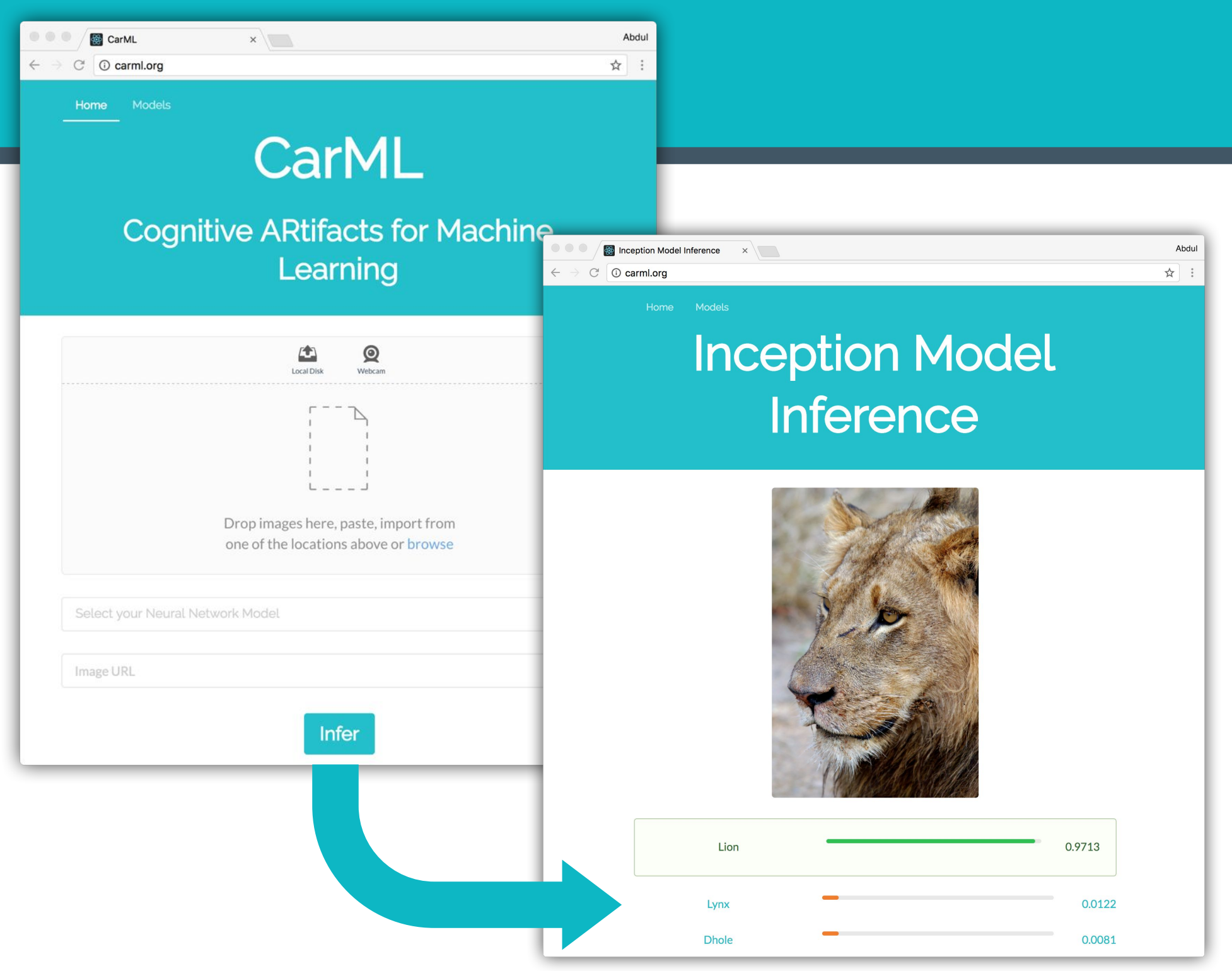
Pipeline and Architecture

1 A user selects the models and inputs. The user then submits them using the CarML web interface.

2 The CarML webserver accepts the user's inputs and forwards the request to the agents capable of evaluating the model.

Evaluation Platform

CarML enables users to upload datasets and select models across frameworks. The CarML platform is accessible through the Web UI or REST API. The figure on the right shows CarML's homepage where users evaluate and experiment with different ML models across inputs.

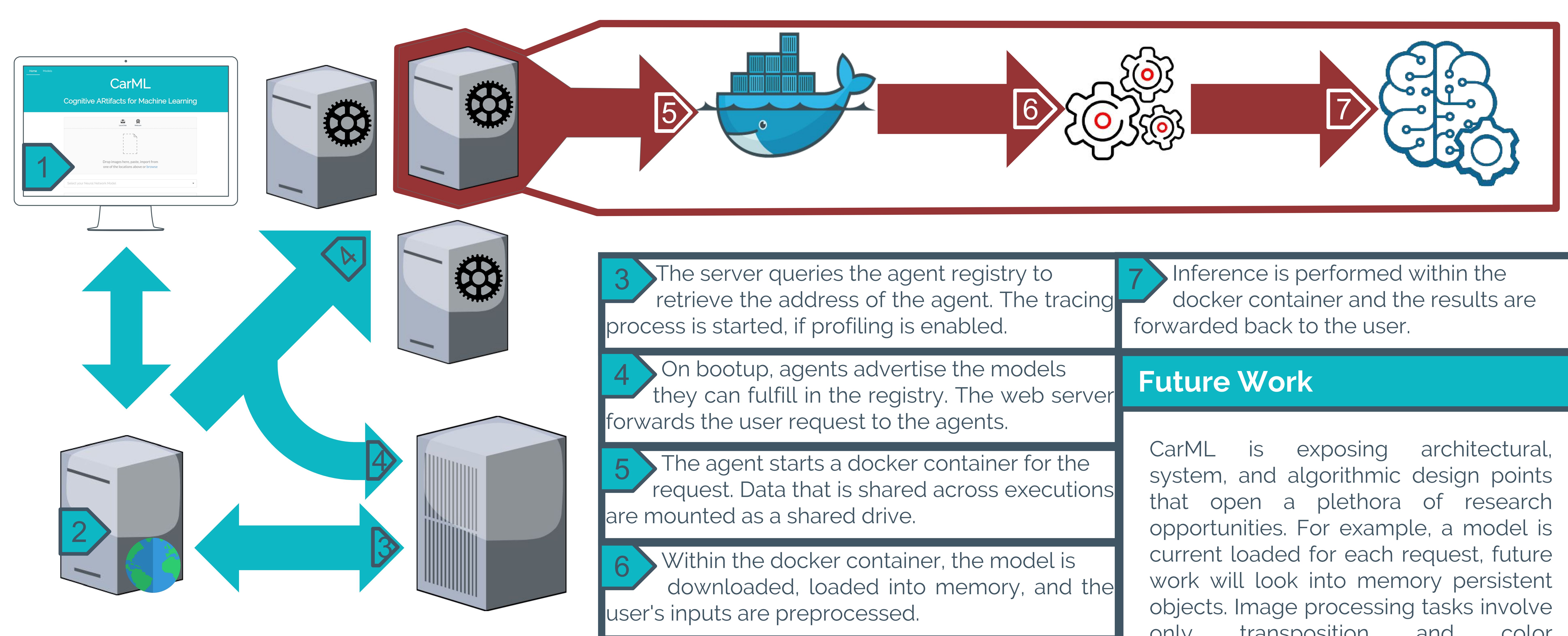


CarML
carml.org
Abdul
Home
Models
CarML
Cognitive ARtifacts for Machine Learning
Local Disk
Webcam
Drop images here, paste, import from one of the locations above or browse
Select your Neural Network Model
Image URL
Infer
Inception Model Inference
carml.org
Abdul
Home
Models
Inception Model Inference

Lion	0.9713
Lynx	0.0122
Dhole	0.0081

Deployment Platform

CarML allows ML/DL developers to expose their models and get feedback. It is designed with different input modalities and support for different machine learning applications. Developers describe their algorithm to CarML with a manifest file. Users can provide the model developer with feedback and suggestions on real workloads



3 The server queries the agent registry to retrieve the address of the agent. The tracing process is started, if profiling is enabled.

4 On bootup, agents advertise the models they can fulfill in the registry. The web server forwards the user request to the agents.

5 The agent starts a docker container for the request. Data that is shared across executions are mounted as a shared drive.

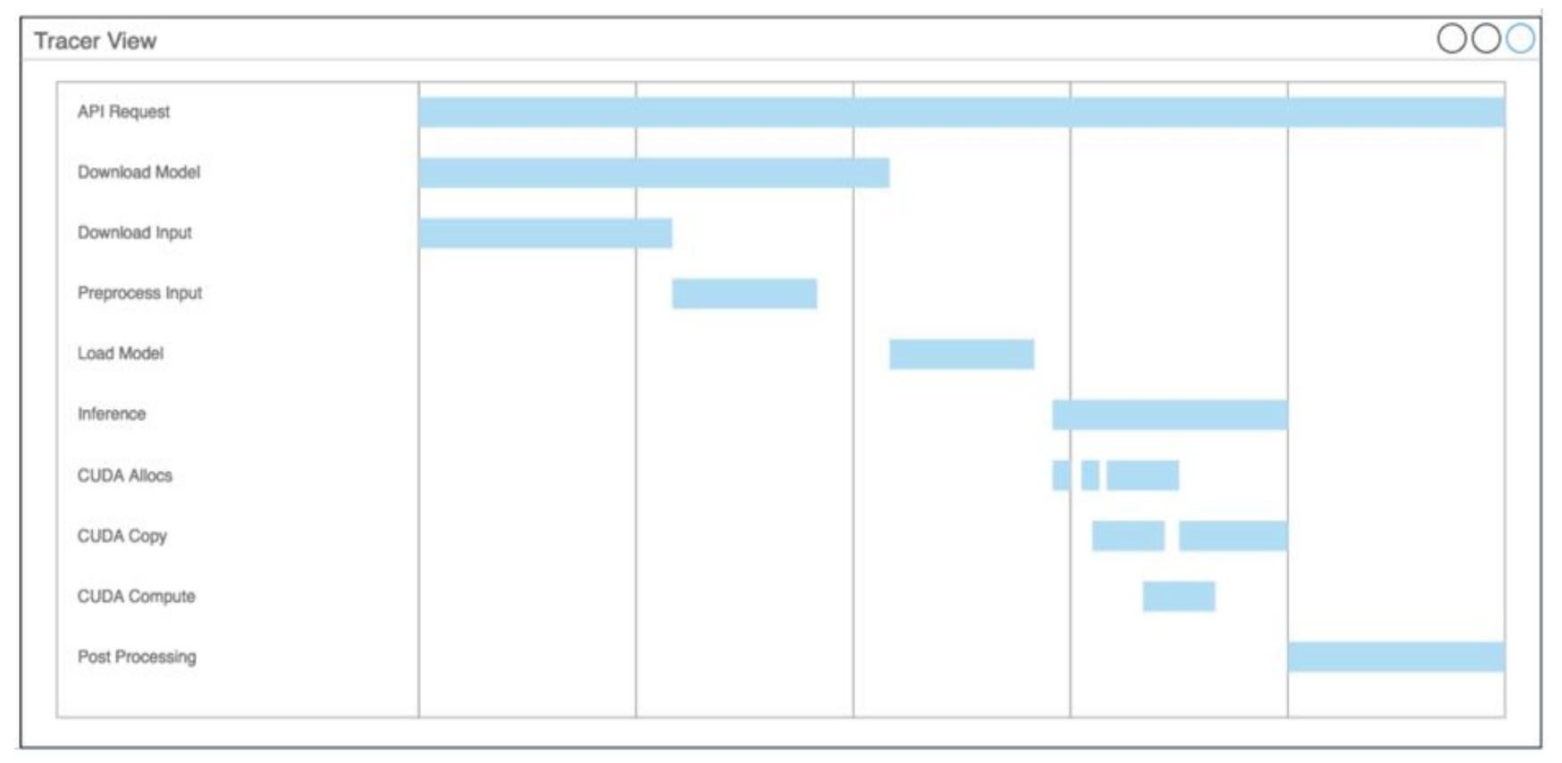
6 Within the docker container, the model is downloaded, loaded into memory, and the user's inputs are preprocessed.

7 Inference is performed within the docker container and the results are forwarded back to the user.

Future Work

CarML is exposing architectural, system, and algorithmic design points that open a plethora of research opportunities. For example, a model is current loaded for each request, future work will look into memory persistent objects. Image processing tasks involve only transposition and color transformation, future work will use Near Memory Acceleration (NMA) technology for acceleration. The current trend of building customized inference processors (e.g. Brainwave from Microsoft, TPU from Google…) can benefit from CarML's evaluation and profiling features. In time, CarML will be the hub to develop, evaluate, and experiment with ML/DL models.

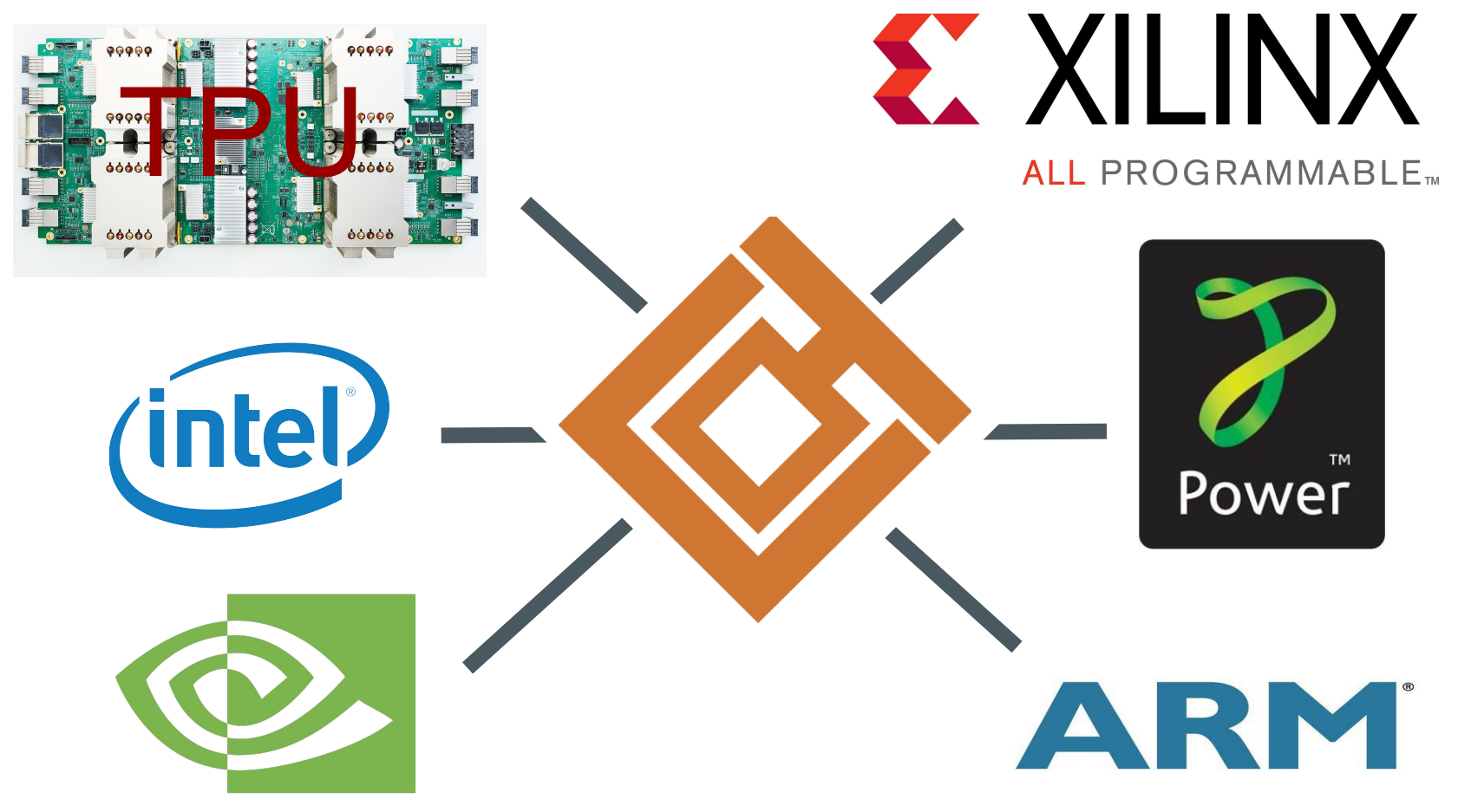
Benchmarking Platform



CarML has distributed tracing capability, allowing tracing information to be captured across server boundaries. The figure on the left shows the trace when profiling is enabled. CarML also provides statistical and per-call function tracing.

Scalability

CarML is a distributed and resilient system where the web server, tracer, and agents span nodes. The figure above shows the CarML architecture. Through its architecture, CarML sports both horizontal and vertical scaling.



rai-project/carml