C3SR Cloud Tools and Services for Heterogeneous Cognitive Computing Systems

Wen-mei Hwu

Professor and Sanders-AMD Chair, ECE, NCSA, CS

University of Illinois at Urbana-Champaign

with

Jinjun Xiong (IBM), Abdul Dakkak, Cheng Li, and Carl Pearson

ECE ILLINOIS | C3SR center for cognitive computing systems research | NCSA | The IMPACT Research Group | NVIDIA | ILLINOIS

Agenda

- Accelerator research at IBM-Illinois C3SR
- RAI
- D4P
- CarML
- Discussions

C3SR Vision
(Center for Cognitive Computing Systems Research)

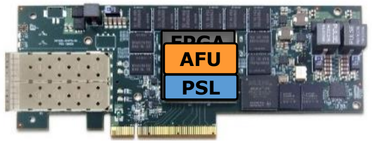
- The rise of cognitive computing has created new opportunities to rethink all the three layers of computing systems– applications, software, and hardware.

- Dramatic enhancement in the efficacy, efficiency and variety of cognitive computing applications can be achieved through innovative system design.

C3SR Experimental Heterogeneous Infrastructure

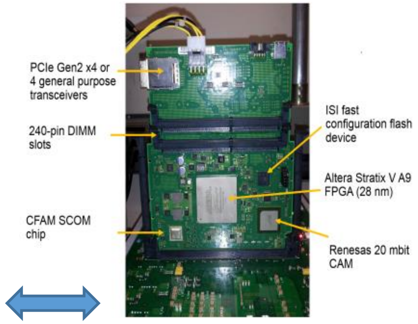
2x P8 Minsky with NVLink Pascal GPUs

DGX-1



FPGA CAPI over PCIe

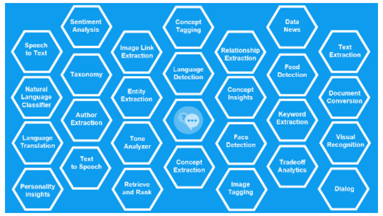
4 x P8 Tuleta (S824L)



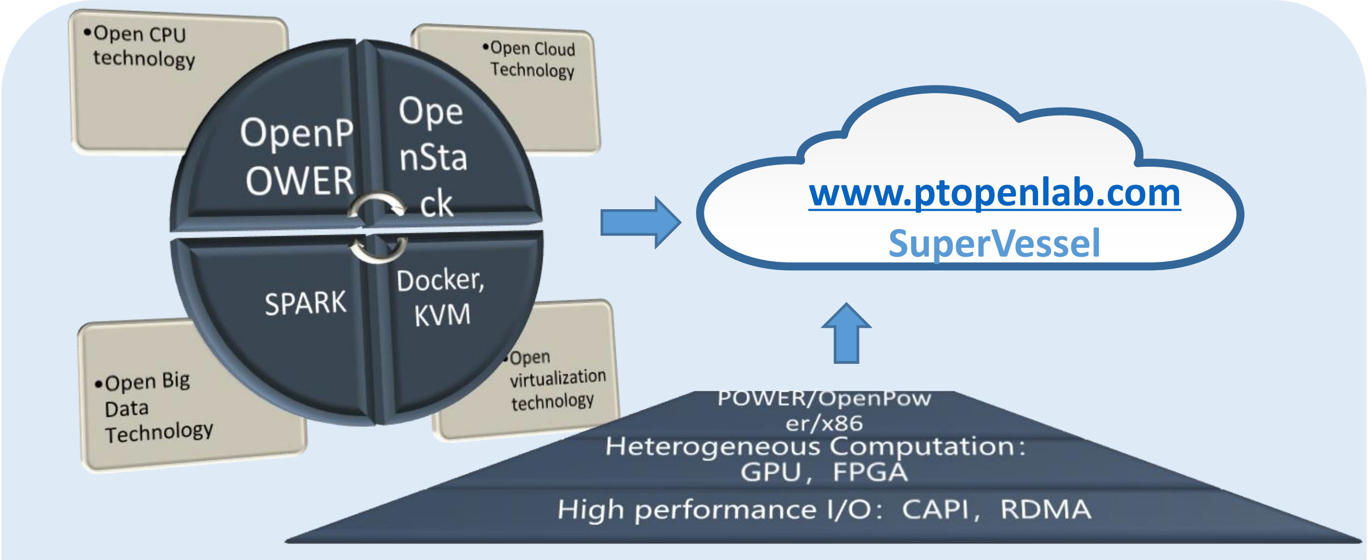
ConTutto over DMI

IBM Bluemix™

IBM SOFTLAYER®



Watson developer cloud



Power9/Volta upgrade in progress!

Accelerator Research Example:

- FPGA accelerated real-time video content recognition with LRCN (Long-term Recurrent Convolutional Network)
  - Achieved 0.04 sec latency: 3x over GPU, 5x over Intel CPU, with x17 lower energy

Internet

A group of young men playing a game of soccer

Front-end (video capturing)
Webcam + Tegra TK1

Back-end (LRCN)
Xilinx FPGA on host server

- More in consideration, including FaceNet, neural machine translation (NMT)

A Common Pattern for Building Cognitive Solutions

Web Service Portal UI

Multi-modality Sensor Input

Video/Image Understanding | Knowledge Representation | Concept Extraction | Formal Assessment | Computational Creativity | NLP/Dialog Systems

OpenPOWER™

NVLINK – Pascal/Volta

OpenCAPI™

FPGA, NMA

Distributed & Heterogeneous Infrastructure

- Applications need to access core services that are optimized for the underlying heterogeneous infrastructure

Agenda

- Accelerator Diversity at IBM-Illinois C3SR
- RAI
- D4P
- CarML
- Discussions

RAI: Easy Use of Accelerators in the Cloud

- Developers download a RAI client binary, which runs on the developer's machine
  - No library dependencies and work on all major OS
- Set up user profile with a secret key to use the RAI service
- **Edit your project locally as you typically do**
- Run the RAI client with pointers to your local project folder, and receive console outputs on your local machine
  - As if you're directly working with a local system with accelerators

https://github.com/rai-project/rai

RAI Demo

```yaml
rai:
  version: 0.2 # this is required
  #  image: gcc:6.3.0
  image: ppc64le/gcc
resources:
  cpu:
    architecture: ppc64le
  network: false
  # gpu:
  #   count: 1
commands:
  build:
    - echo "Building project"
    - gcc /src/main.c
    - ./a.out

```

Submission Spec

```c

#include "stdio.h"

int main() {
    printf("Hello Universe!!\n");
    return 0;
}
```

User Program

Output

```
✱ Checking your athentication credentials.
✱ Preparing your project directory for upload.
✱ Uploading your project directory. This may take a few minutes.
 358 B / 358 B  100.00% 5.23 KiB/s 0s
✱ Folder uploaded. Server is now processing your submission.
✱ Your job request has been posted to the queue.
✱ Server has accepted your job submission and started to configure the container.
✱ Downloading your code.
✱ Using ppc64le/gcc as container image.
✱ Starting container.
✱ Running echo "Building project"
Building project
✱ Running gcc /src/main.c
✱ Running ./a.out
Hello Universe!!
✱✱ The build folder has been uploaded to http://s3.amazonaws.com/files.rai-project.com/userdata/build-377d8ae0-64da-441c-80fb-bff5e717e13f.tar.tar.gz. The data will be present for only a short duration of time.
✱ Server has ended your request.
```

https://asciinema.org/a/6k5e96itnqu6ekbji60c3kgy4

RAI: Current Use (and X86 too)

- We have been using RAI extensively for teaching at UIUC
  - ~270 students registered the UIUC's GPU Programming Class (ECE408/CS483)
  - ~150 students registered the UIUC's GPU Algorithm Class (ECE508/CS508)
  - ~100 students all around the world attending the Programming and Tuning Massively Parallel Systems (PUMPS) summer school
- Supported tasks such as
  - Students to develop a CUDA version of a CNN
  - Students to use system profiling tools to identify performance bottlenecks
  - Students allowed for repeated submissions in a competition
  - Teachers to grade repeated submissions automatically
- System has to be scalable and elastic (from 1 to 20 AWS instances!)

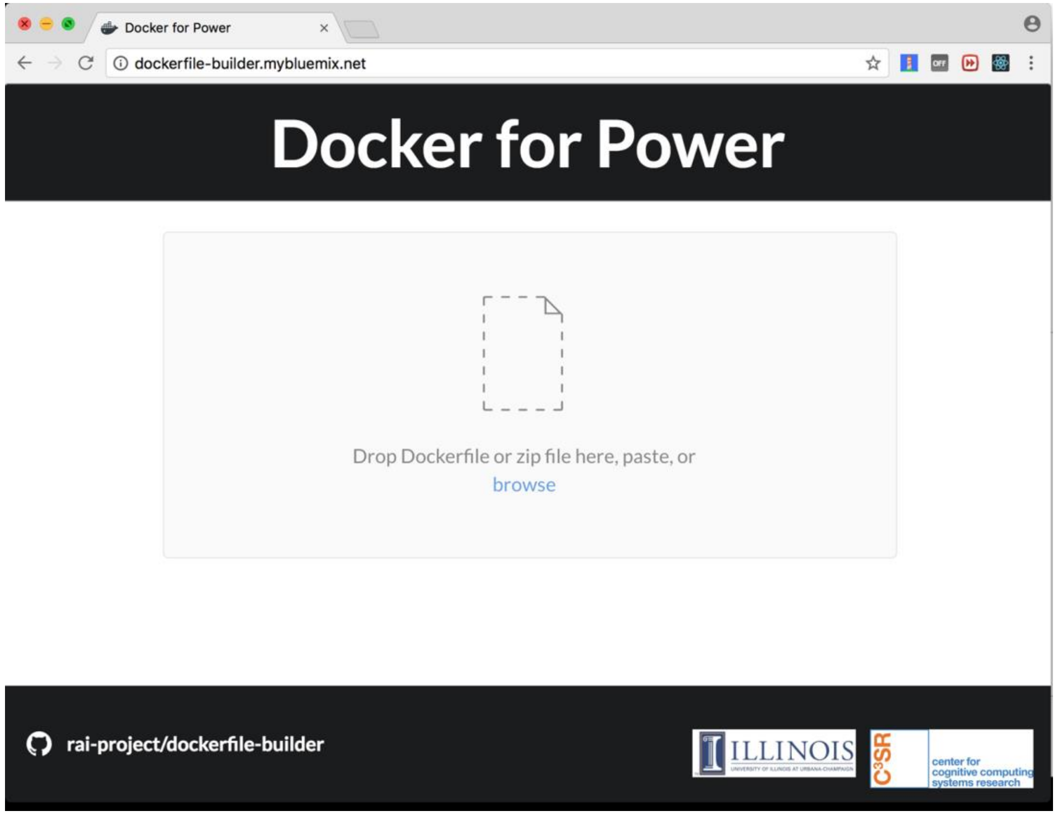
Agenda

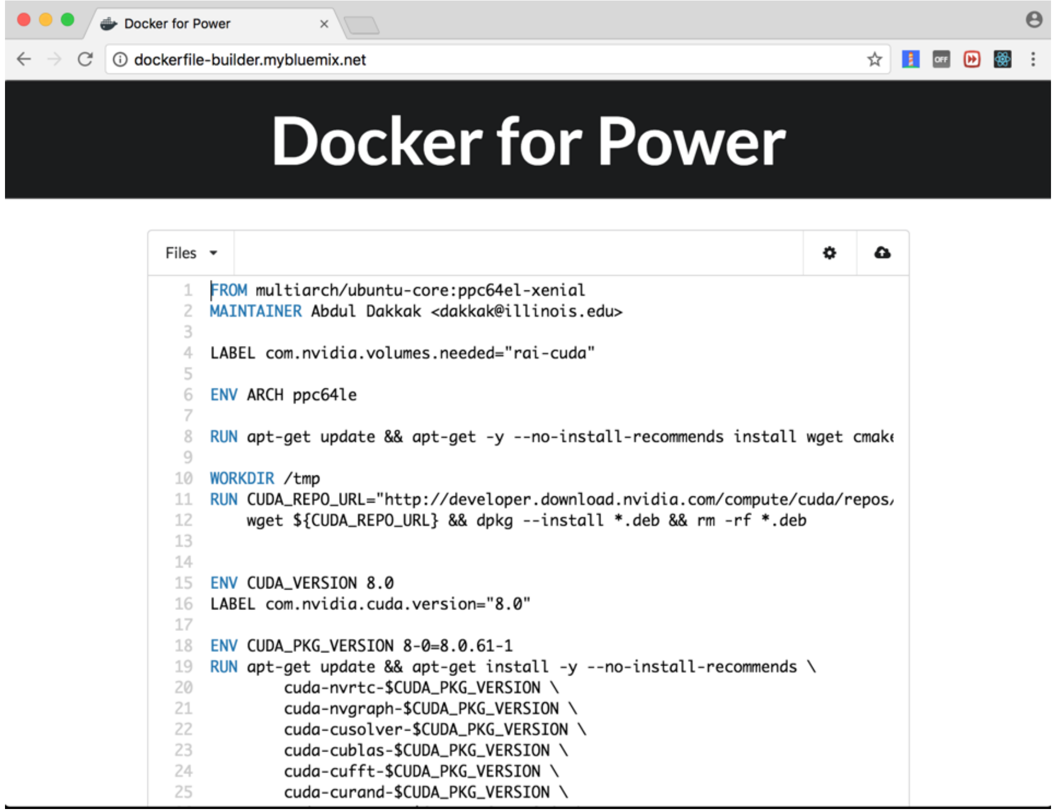
- Accelerator Diversity at IBM-Illinois C3SR
- RAI
- D4P
- CarML
- Discussions

D4P: Docker for POWER

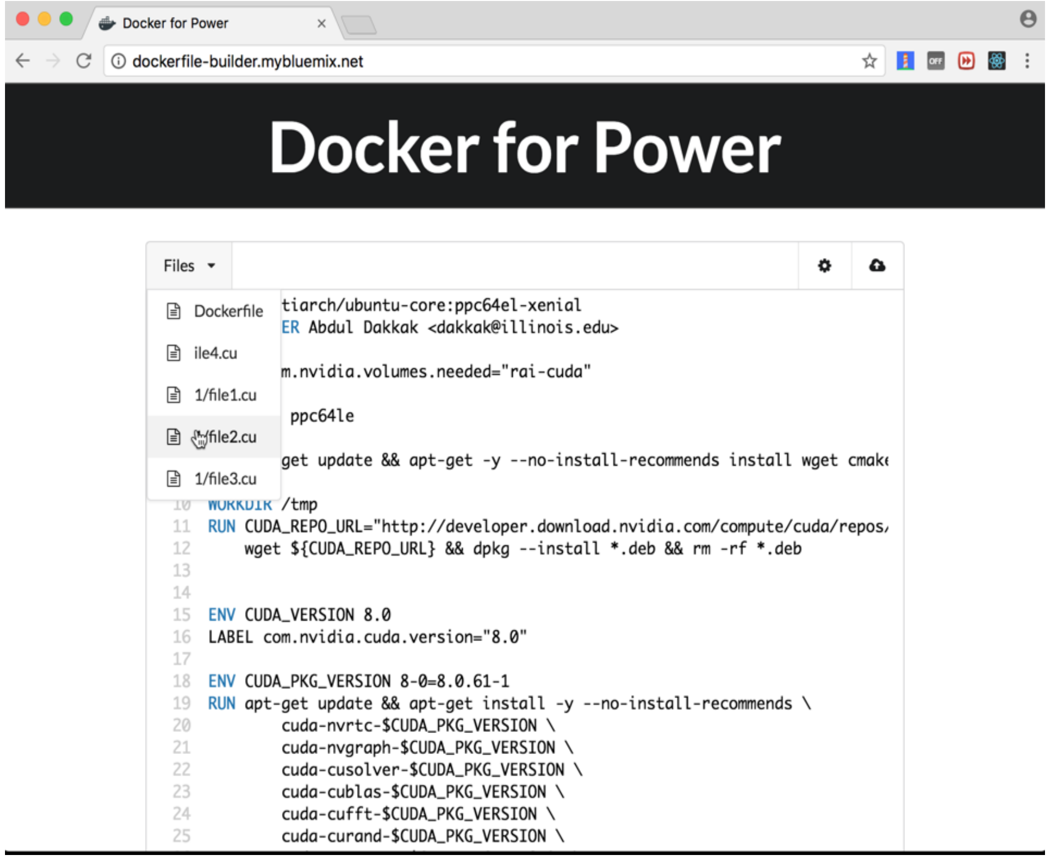
- Objectives
  - Extend the POWER Docker ecosystem by making it possible to build images without direct access to POWER hardware
  - Make building and deploying POWER Docker images easy for the developers
  - A home for POWER Docker containers
- D4P provides
  - A cloud-based service for authoring and publishing POWER Docker images
  - An API interface for easy integration with any dev/ops pipelines (e.g., for building POWER-compatible packages)
  - A fast increasing collection of Docker images for POWER/accelerator-compatible packages

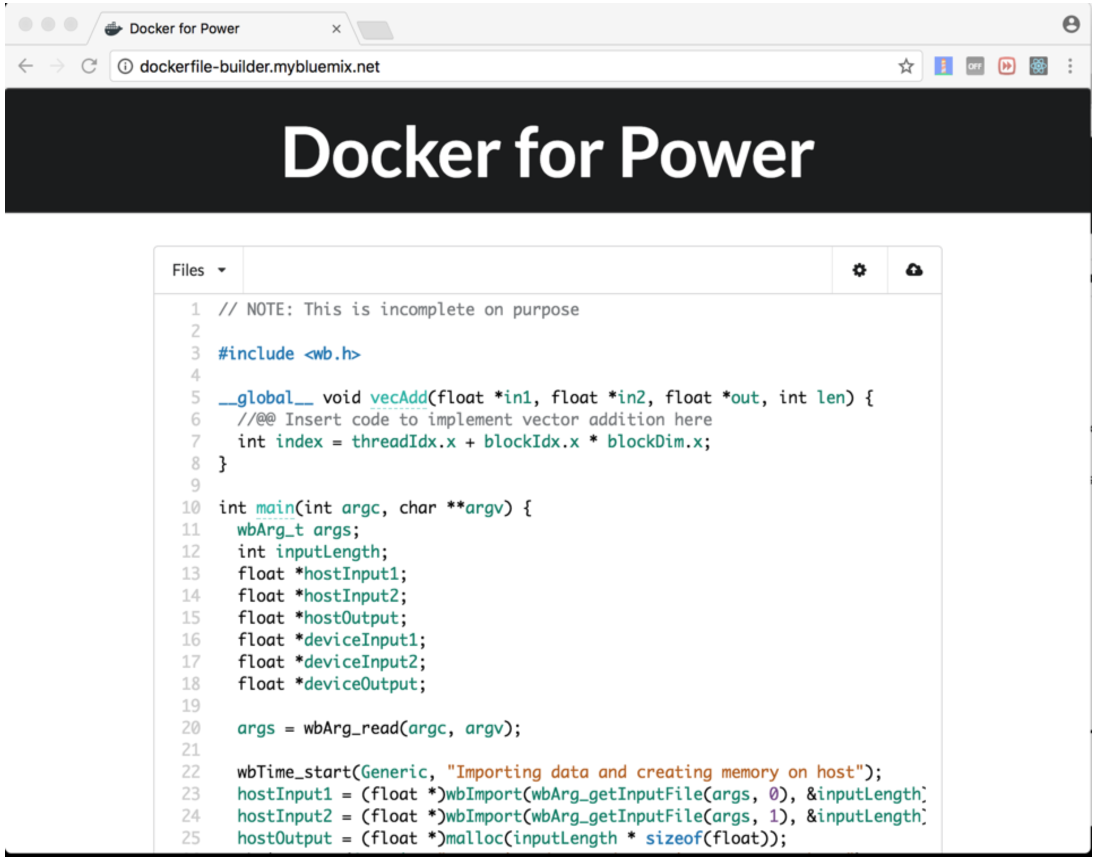
D4P demo


Docker for Power
dockerfile-builder.mybluemix.net
Docker for Power
Drop Dockerfile or zip file here, paste, or
browse
rai-project/dockerfile-builder
ILLINOIS
C3SR
center for cognitive computing systems research

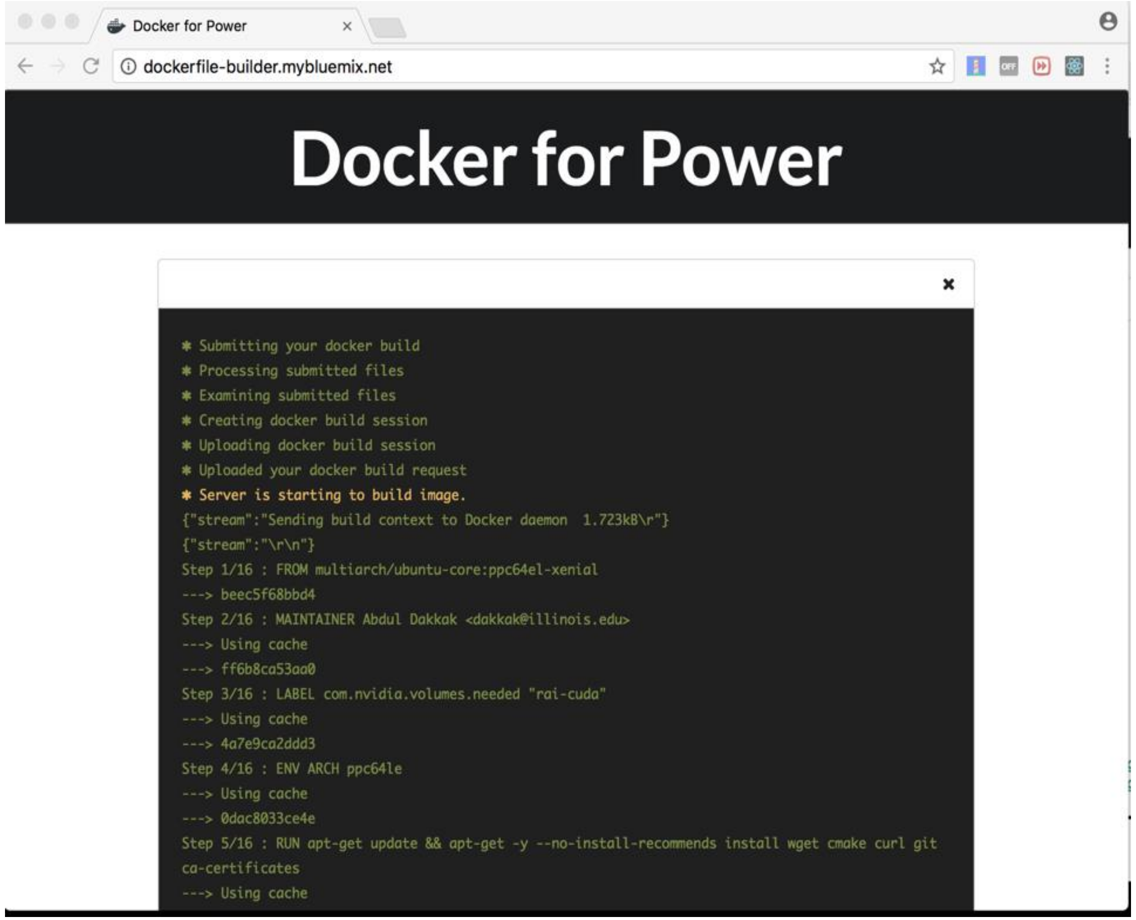

Docker for Power
dockerfile-builder.mybluemix.net
Docker for Power
Files
FROM multiarch/ubuntu-core:ppc64el-xenial
MAINTAINER Abdul Dakkak <dakkak@illinois.edu>
LABEL com.nvidia.volumes.needed="rai-cuda"
ENV ARCH ppc64le
RUN apt-get update && apt-get -y --no-install-recommends install wget cmake
WORKDIR /tmp
RUN CUDA_REPO_URL="http://developer.download.nvidia.com/compute/cuda/repos/
wget ${CUDA_REPO_URL} && dpkg --install *.deb && rm -rf *.deb
ENV CUDA_VERSION 8.0
LABEL com.nvidia.cuda.version="8.0"
ENV CUDA_PKG_VERSION 8-0=8.0.61-1
RUN apt-get update && apt-get install -y --no-install-recommends \
cuda-nvrtc-$CUDA_PKG_VERSION \
cuda-nvgraph-$CUDA_PKG_VERSION \
cuda-cusolver-$CUDA_PKG_VERSION \
cuda-cublas-$CUDA_PKG_VERSION \
cuda-cufft-$CUDA_PKG_VERSION \
cuda-curand-$CUDA_PKG_VERSION \

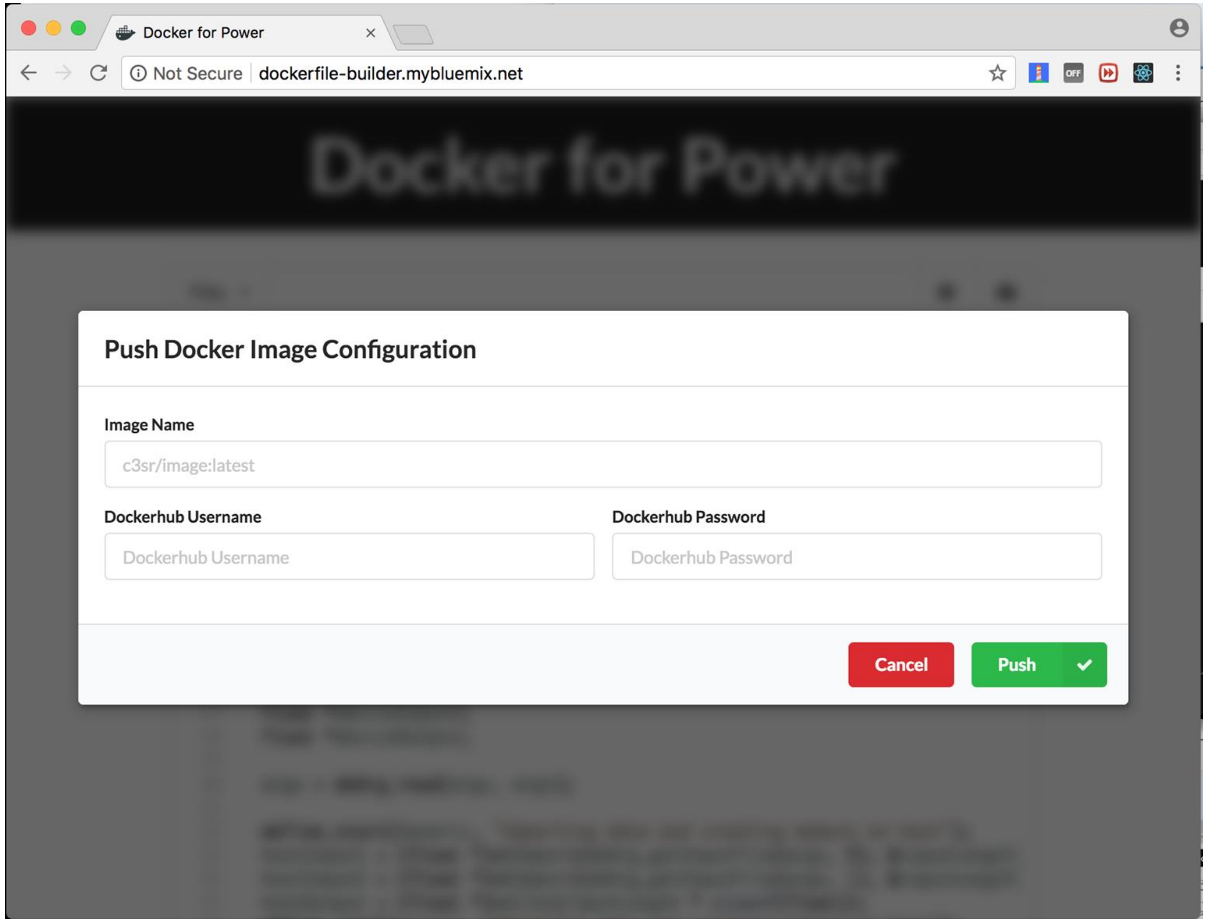
D4P demo: authoring and editing



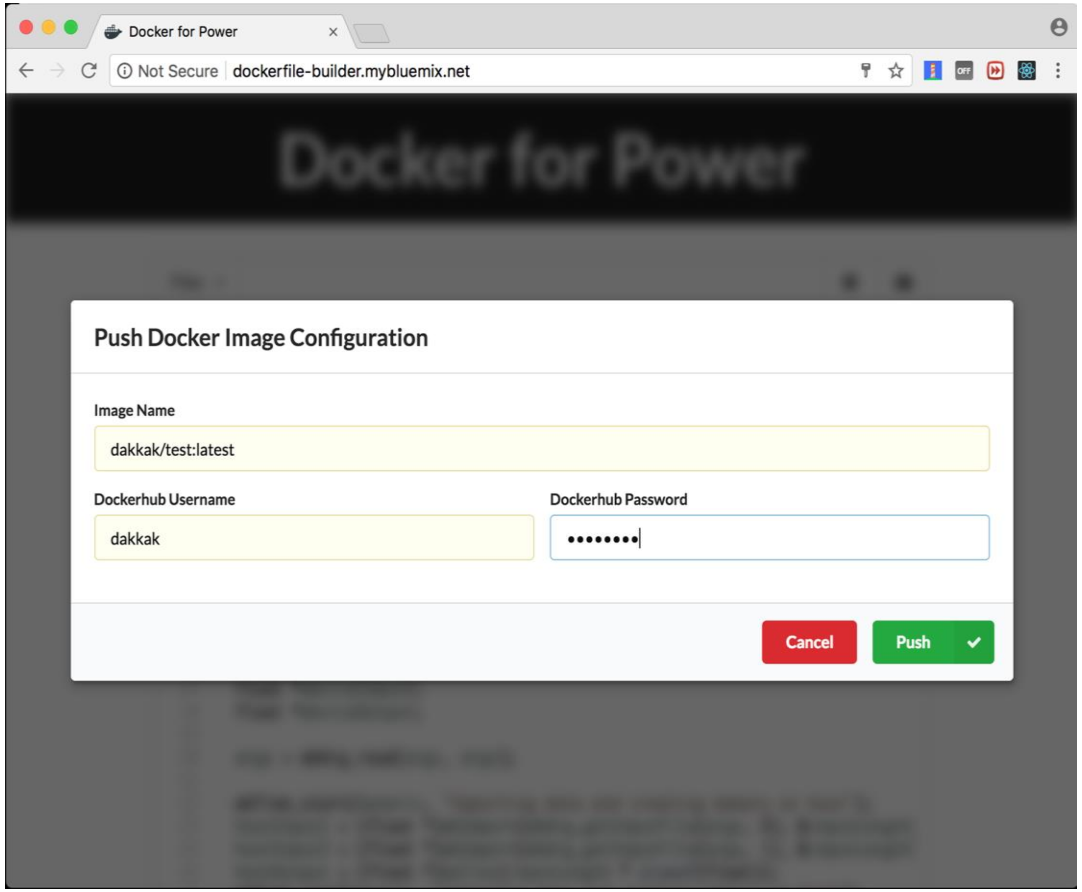


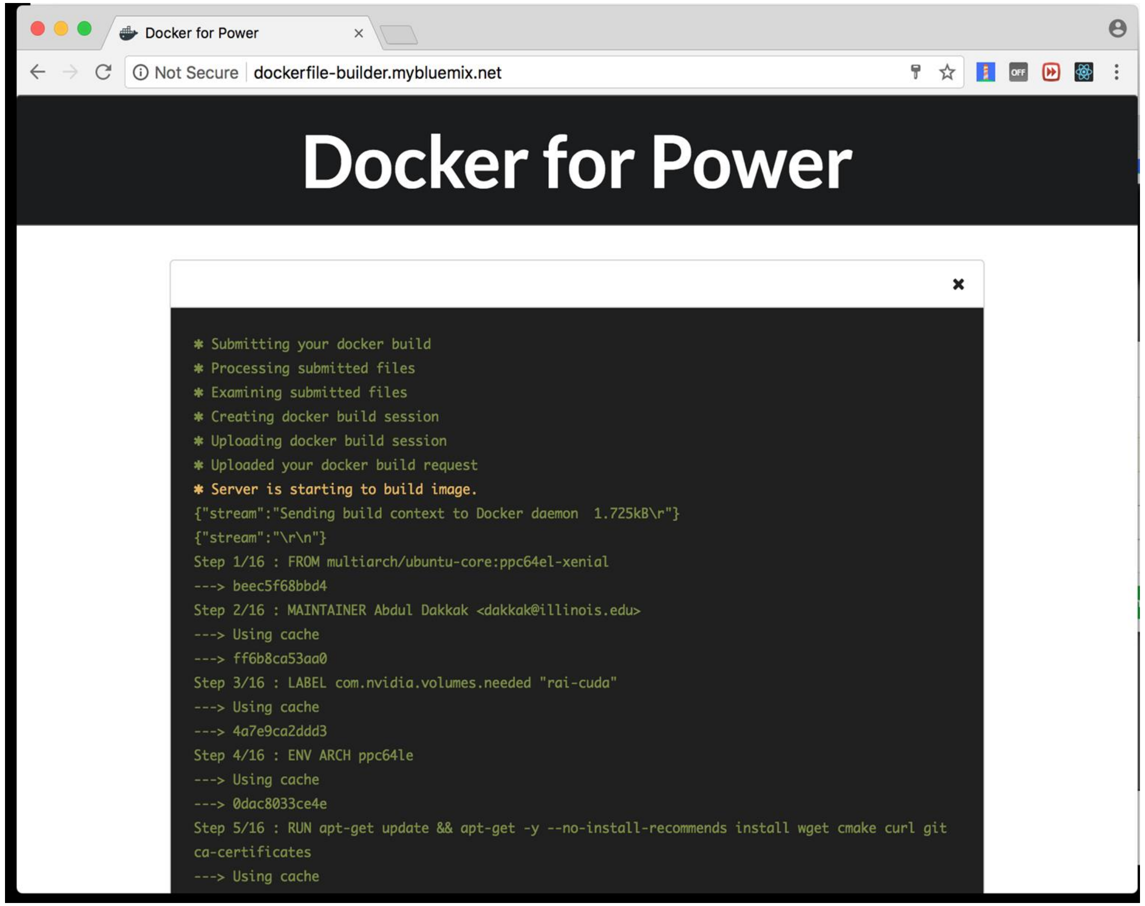
D4P demo: building and publishing





D4P: publishing docker images to docker hub





D4P: a hub for POWER Docker images

Images About

Docker for Power

Name	Dockerfile	Published
c3sr/bonita:7.4.2		Yes
c3sr/celery:4.0.2		Yes
c3sr/consul:0.8.1		Yes
c3sr/crate:1.0.5		Yes
c3sr/joomla:3.6.5		Yes
c3sr/kaazing-gateway:5.5.0		Yes
c3sr/lynx:latest		Yes
c3sr/proj.4:latest		Yes
c3sr/pryramid_mako:latest		Yes
c3sr/python-stripe:latest		Yes
c3sr/vincent:latest		Yes
c3sr/headers_workaround:latest		Yes

Agenda

- Accelerator Diversity at IBM-Illinois C3SR
- RAI
- D4P
- CarML
- Discussions

ML/DL ecosystem: status-quo

- Diverse models
  - New DL models are popping up almost everyday around the world on arXiv/github
- Diverse frameworks
  - Theano, Caffee, Tensorflow, Torch, MXNET, Chainer…
- Diverse hardware infrastructures
  - X86, POWER, GPUs, FPGAs, accelerators…

A platform allowing model users to easily evaluate and consume ML models and algorithms

- Try different ML models with a click
- Run different ML models on user provided data
- Validate ML models performance / accuracy
- Benchmark HW impacts on ML models in terms of performance, energy & cost

A deployment platform for ML model researchers to promote their research and receive timely feedback

- Easy to publish a new ML model for anyone to try it
  - Users can reproduce results
  - Model variety with different input / output modalities (text, voice, images etc.)
  - Framework variety with different packages (Caffee, Tensorflow, Torch etc.)

- Receive feedback on test cases where models break (e.g., unseen cases)

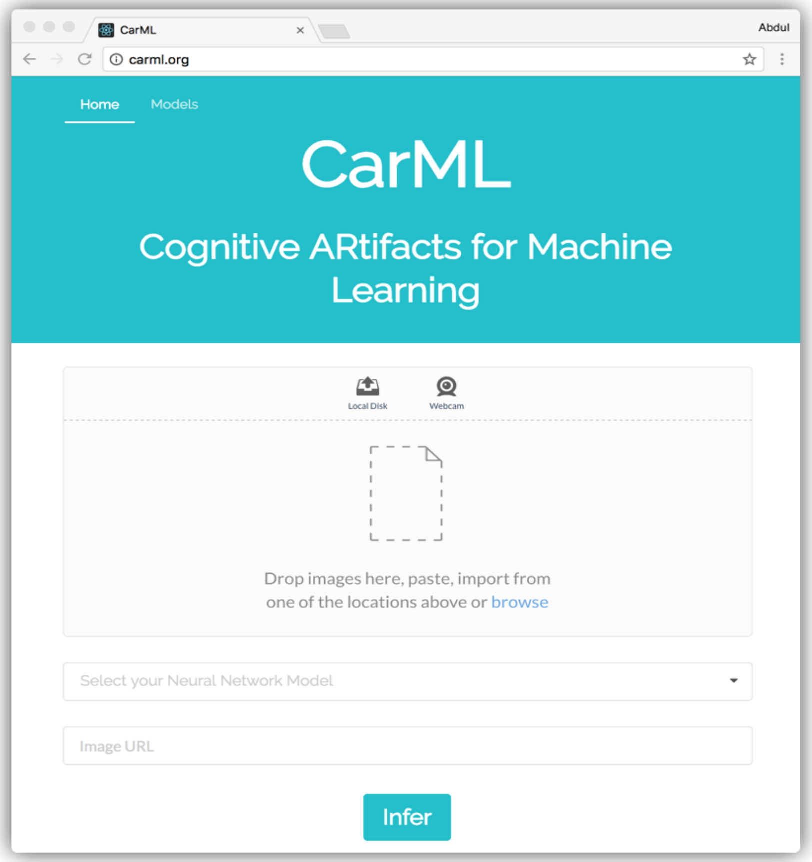
- Easy to benchmark against peers' results (scoreboards)

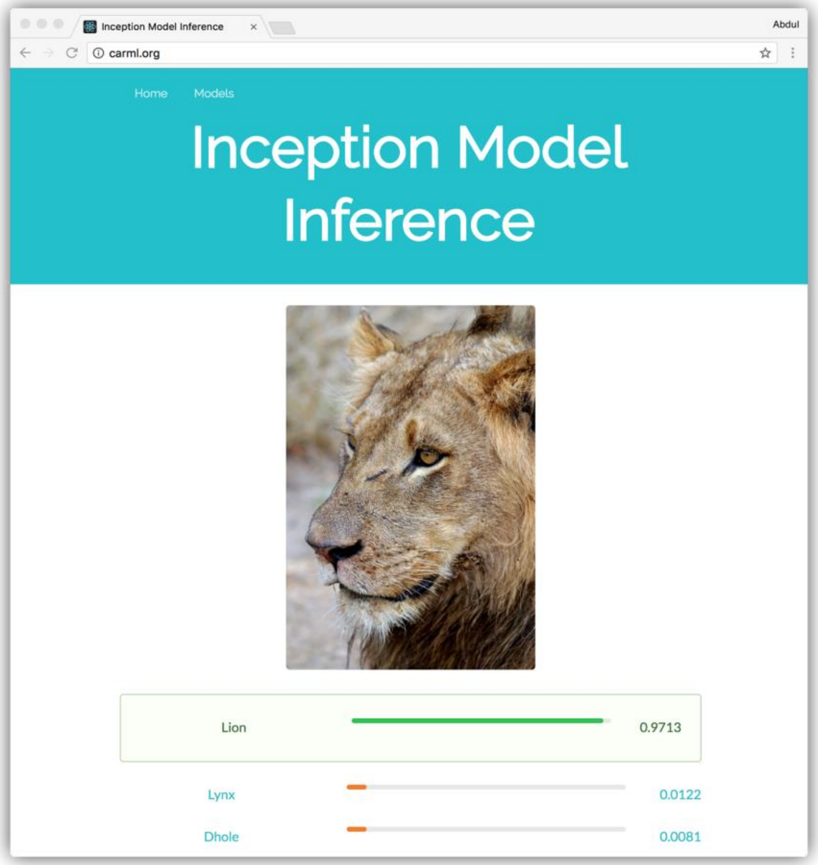
A workload characterization platform to understand system bottlenecks for ML workloads

- All major frameworks, data sets, models available
- Provide distributed tracing and monitoring capabilities
- Support different HW infrastructures
- Allow easy integration of new HW innovations

CarML: prototype demo

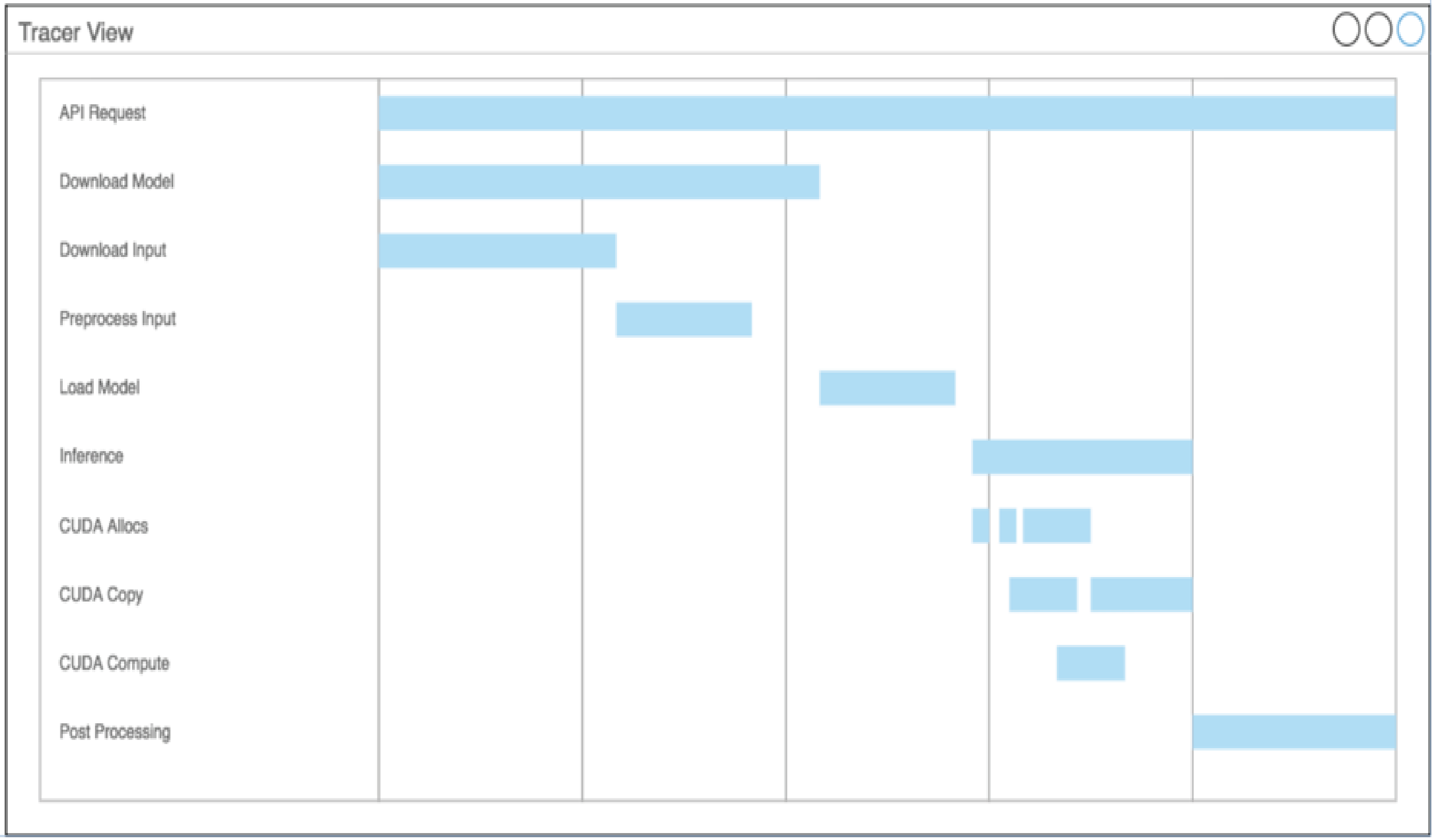
- www.carml.org





CarML: end-to-end system tracing demo

- 52.44.160.49:9411



CarML: an open platform to answer those challenges

- Deploy and benchmark machine learning frameworks and models across hardware infrastructures, through a common interface
  - An experimentation platform for ML users
  - A deployment platform for ML developers
  - A benchmarking platform for systems architects
- A distributed and resilient system where the web server, registry, tracer, and agents can all scale either horizontally or vertically

C3SR

center for
cognitive computing
systems research

Dr. Jinjun Xiong (IBM)

Abdul Dakkak

Cheng Li

Carl Pearson

Thanks!


Suma Bhat


Minh Do


Deming Chen


Julia Hockenmaier


Wen-mei Hwu


Nam Sung Kim


Dan Roth


Rakesh Nagi

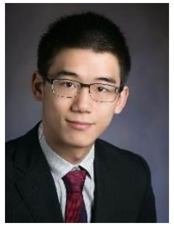
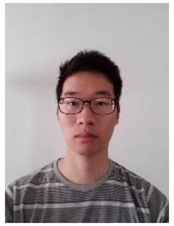

Lav Varshney



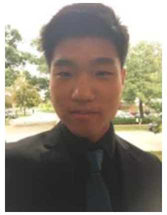






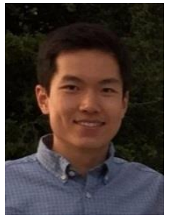


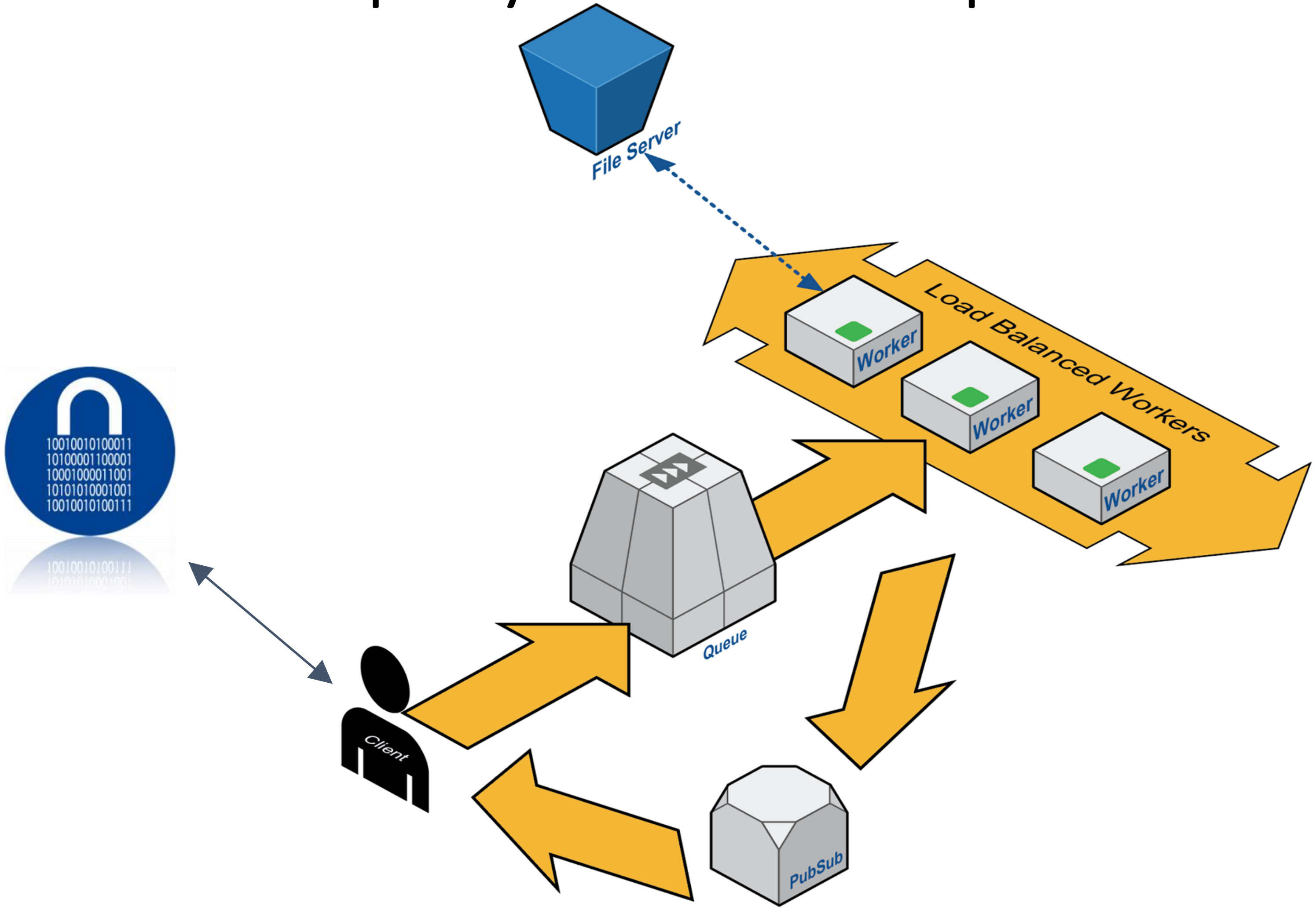




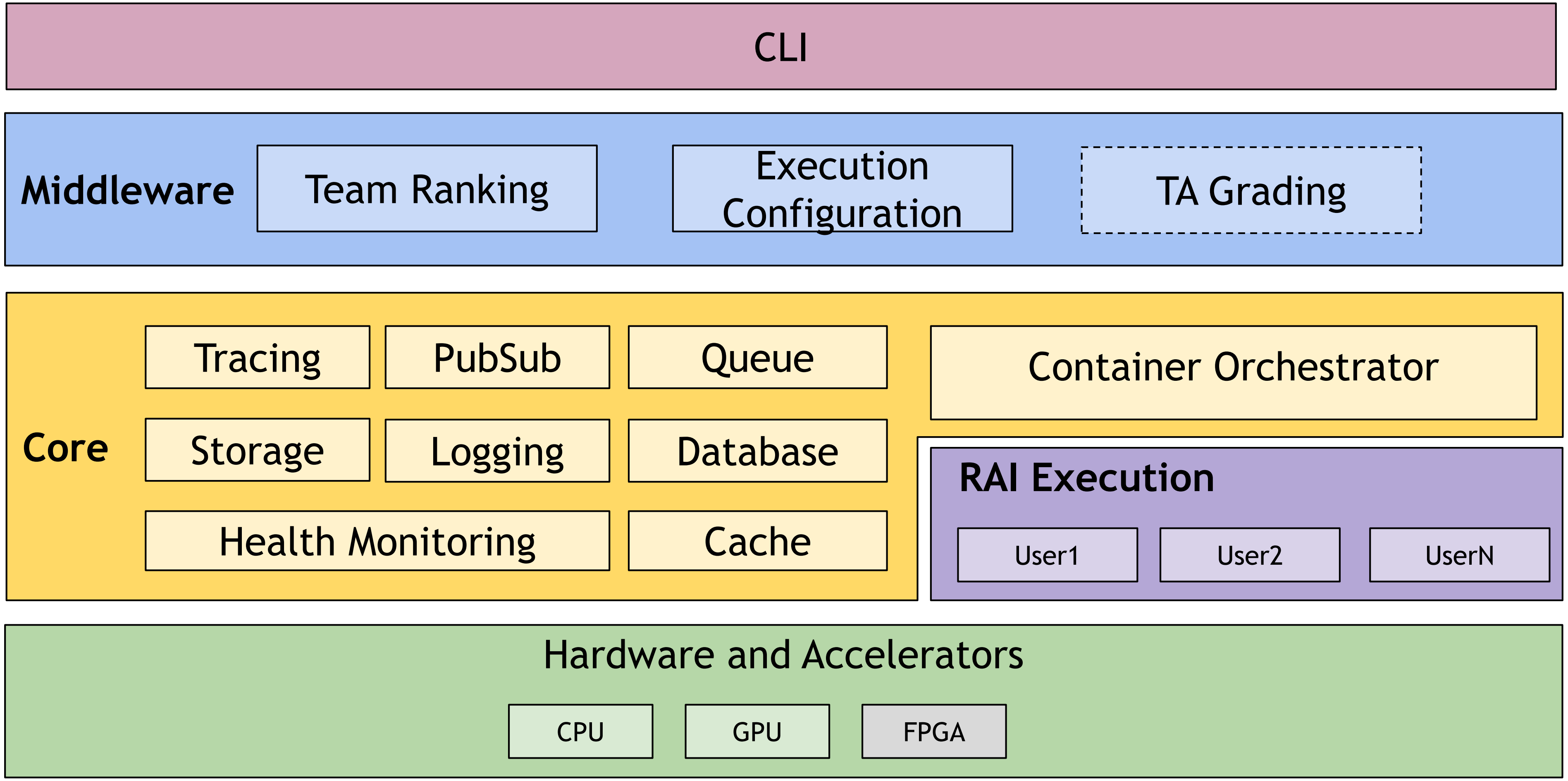


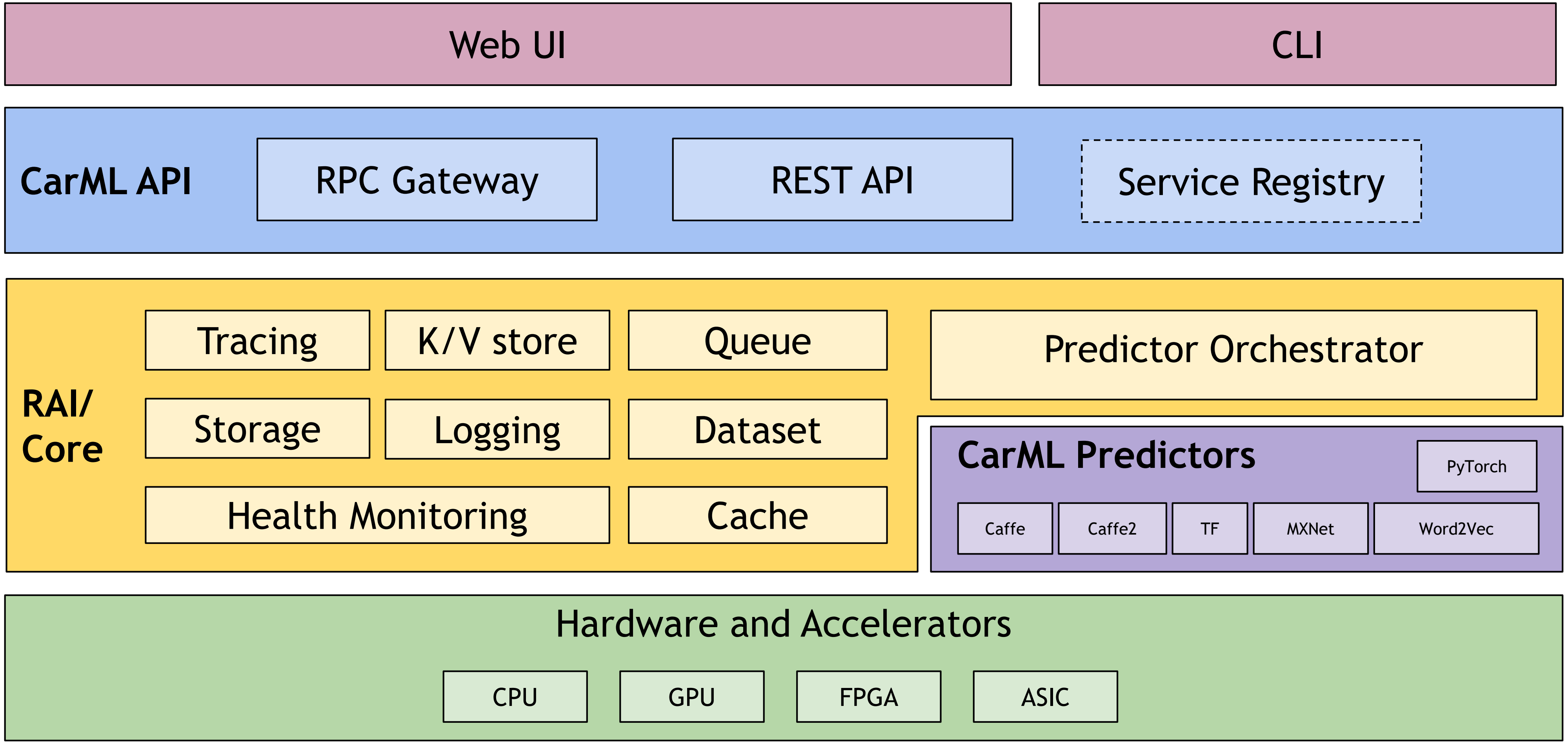
RAI’s current deployment setup



RAI architecture with reusable components

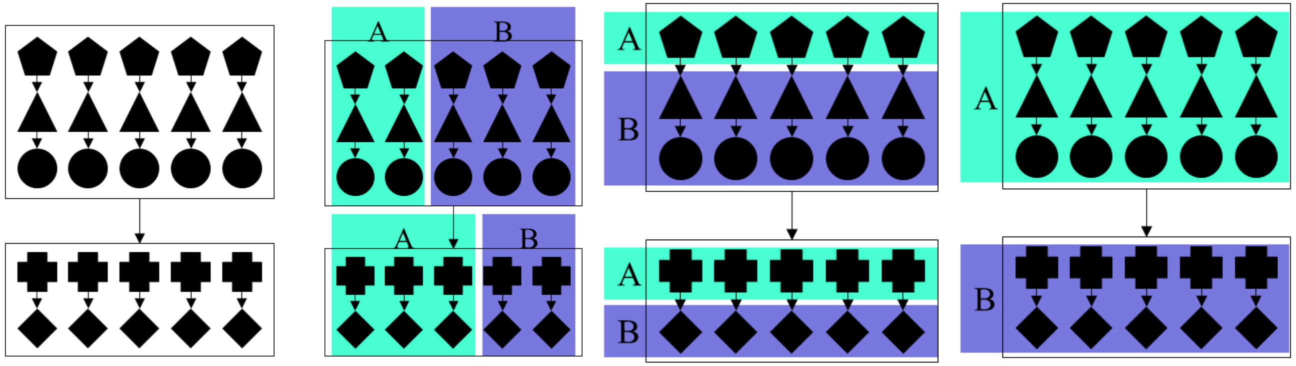


CarML architecture: built on RAI



Cognitive benchmarks optimized for POWER

- Motivation: demonstrate the value of a well-balanced CPU + accelerators design for many important workloads
- Chai (Collaborative Heterogeneous Applications for Integrated-architecture)
  - Identified a set of common collaborative computation patterns
  - Demonstrated benefits of having CPU + accelerators for those patterns
    - Primary on AMD Kaveri A10-7850K APU
  - Open sourced a set of benchmarks to evaluate various CPU + accelerators architectures



Collaboration Pattern		Short Name	Benchmark
Data Partitioning		BS	Bézier Surface
		CEDD	Canny Edge Detection
		HSTI	Image Histogram (Input Partitioning)
		HSTO	Image Histogram (Output Partitioning)
		PAD	Padding
		RSCD	Random Sample Consensus
		SC	Stream Compaction
		TRNS	In-place Transposition
Task Partitioning	Fine-grain	RSCT	Random Sample Consensus
		TQ	Task Queue System (Synthetic)
		TQH	Task Queue System (Histogram)
	Coarse-grain	BFS	Breadth-First Search
		CEDT	Canny Edge Detection
		SSSP	Single-Source Shortest Path

- On-going: add more cognitive-related benchmarks + release an optimized version for POWER systems

Power Accelerator Ecosystems: status-quo

Learners

The POWER Minsky with NVLink GPUs (or CAPI FPGA) is so cool. Can I learn how to program them?

I'm a big fan of accelerator technologies.
How can I educate my students/peers about it at scale?

Educators

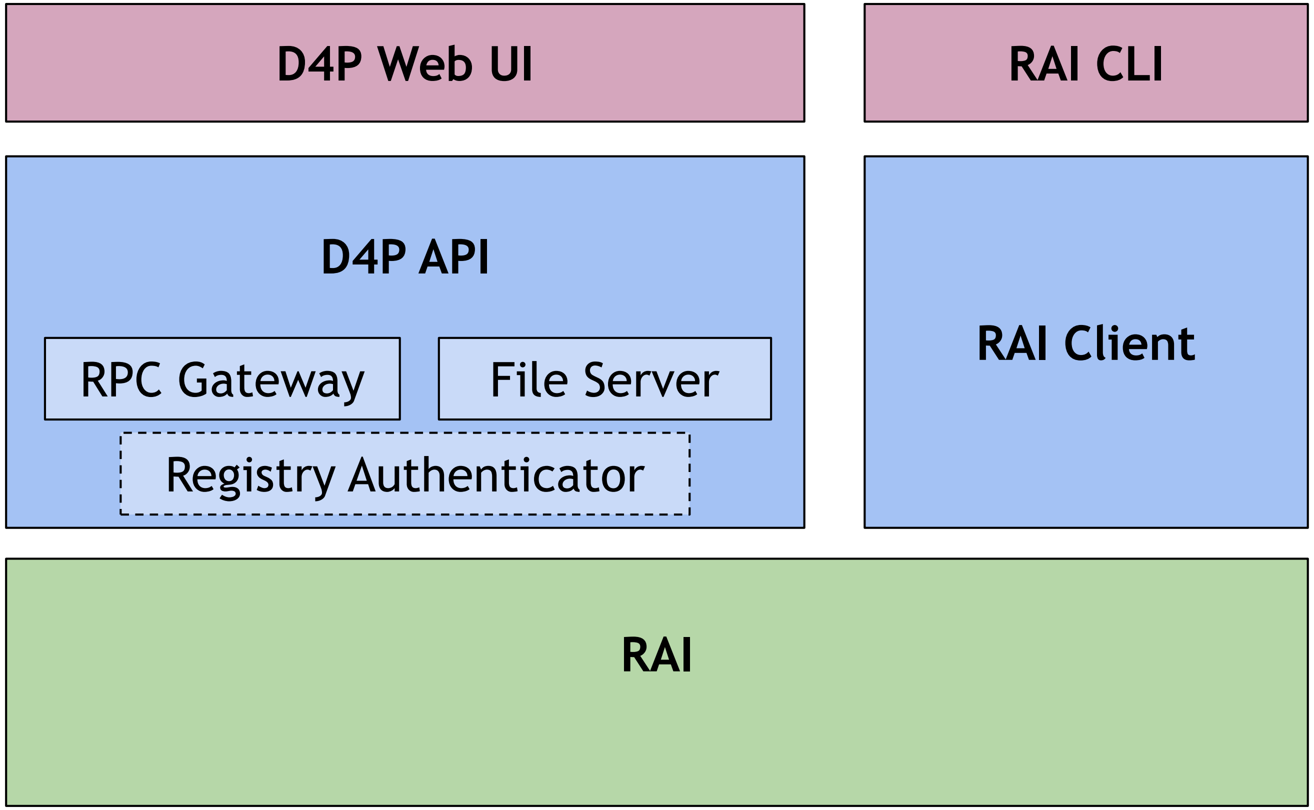
Developers

I have a great Open Source project.
How can I make use of accelerators in the cloud?

RAI: built on Many Existing Open Source Projects

Services	Available Backends
Authentication	Secret, **Auth0**
Queue	NSQ, **SQS**, Redis, Kafka, NATS
Database	RethinkDB, MongoDB, MySQL, Postgres, SQLite, ...
Registry	Etcd, Consul, BoltDB, Zookeeper
Config	**Yaml**, Toml, JSON, Environment
PubSub	EC, **Redis**, GCP, NATS, SNS
Trace	XRay, Zipkin, StackDriver, Jaeger
Logger	**StackDriver**, **JournalD**, Syslog, Kinesis
Store	**S3**, Minio, Memfs, LMDB
Container	**Docker**
Serializer	BSON, **JSON**, YAML, JSONPB, Python Pickle

D4P Architecture: built on top of RAI



ML/DL ecosystem personas: users

Business innovator

There are so many cool DL models.
Which one will work (or will there be one working) for my data?

I just heard of a new wonderful DL model published on arXiv/github. Can it really achieve that impressive results?

ML enthusiasts

IT support for business

What hardware (performance, energy, cost) should I buy to support the desired business logic for adopting DL models / algorithms?

CarML: model researchers

I just published such a wonderful DL model.
How can I let the world to try it without me providing too much support (documentation)?

Model researcher

I heard people are using my DL models. Does it work all the time? If not, what can I do to improve my model for interesting scenarios?

Model researcher

How does my model compare against the latest models that are constantly popping up from almost everywhere?

Model researcher

ML/DL ecosystem personas: system researchers

AI is the future, and ML/DL will be a key workload.
How can I characterize those workloads (with so many models and frameworks) on my HW systems?

System researcher

I have designed a new wonderful HW system.
Will it work seamlessly and wonderfully for those existing ML/DL models?

System researcher

People are complaining my systems not performing for their DL models. How can I easily repeat the same experiment as them?

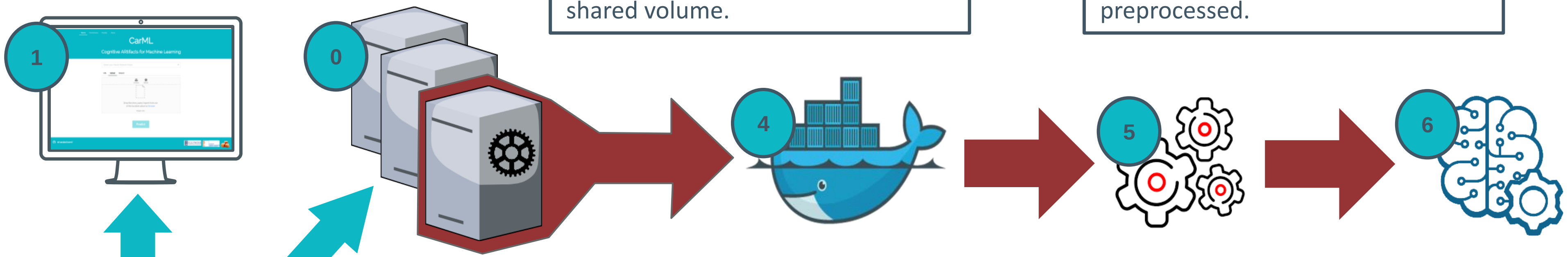
System researcher

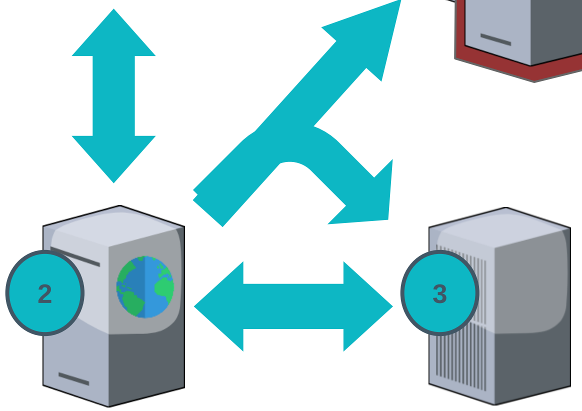
CarML: workflow explained

1 A use selects models and inputs through web UI or API.

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

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





0 On bootup, agents advertise the models in the registry. The web server forwards the user request to the agents capable of evaluating the model

6 Inference is performed within the docker container and the results are sent back to the user.

2 The web server accepts the user inputs and interacts with registry, tracing, and agents

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