

R0Cm and Hopsworks for end-to-end deep learning pipelines



Jim Dowling, CEO, Logical Clocks
Ajit Mathews, CVP ML Software Eng, AMD

17 October 2019
O'Reilly AI, London



Great Hedge of India

- East India Company was one of the industrial world's first monopolies.
- They assembled a thorny hedge (not a wall!) spanning India.
- You paid customs duty to bring salt over the wall (sorry, hedge).

In 2019, not all graphics cards are allowed to be used in a Data Center.

Monopolies are not good for deep learning!



Nvidia™ 2080Ti vs AMD Radeon™ VII: ResNet-50

Nvidia™ 2080Ti

Memory: 11GB

TensorFlow 1.12

CUDA 10.0.130, cuDNN 7.4.1

Model: **RESNET-50**

Dataset: imagenet (synthetic)

FP32 total images/sec: ~**322**

FP16 total images/sec: ~**560**

References:

<https://lambdalabs.com/blog/2080-ti-deep-learning-benchmarks/>

<https://www.phoronix.com/scan.php?page=article&item=nvidia-rtx2080ti-tensorflow&num=2>

AMD Radeon™ VII

Memory: 16 GB

TensorFlow 1.14

ROCm: 2.7

Model: **RESNET-50**

Dataset: imagenet (synthetic)

FP32 total images/sec: ~**316**

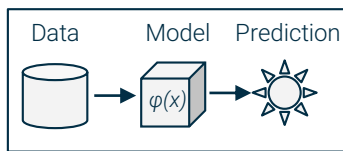
FP16 total images/sec: ~**421**

Reference:

<https://github.com/ROCmSoftwarePlatform/tensorflow-upstream/issues/173>

Hopsworks hides the Complexity of Deep Learning

Hopsworks
Feature Store



Hopsworks
REST API

Datasources

Hopsworks

The Platform for Data Intensive AI
-
Machine Learning, Deep Learning &
Model serving

Applications

API

Dashboards

Hopsworks

Datasources

Batch

Streaming

Distributed
Machine Learning
&
Deep Learning

Serving

Applications

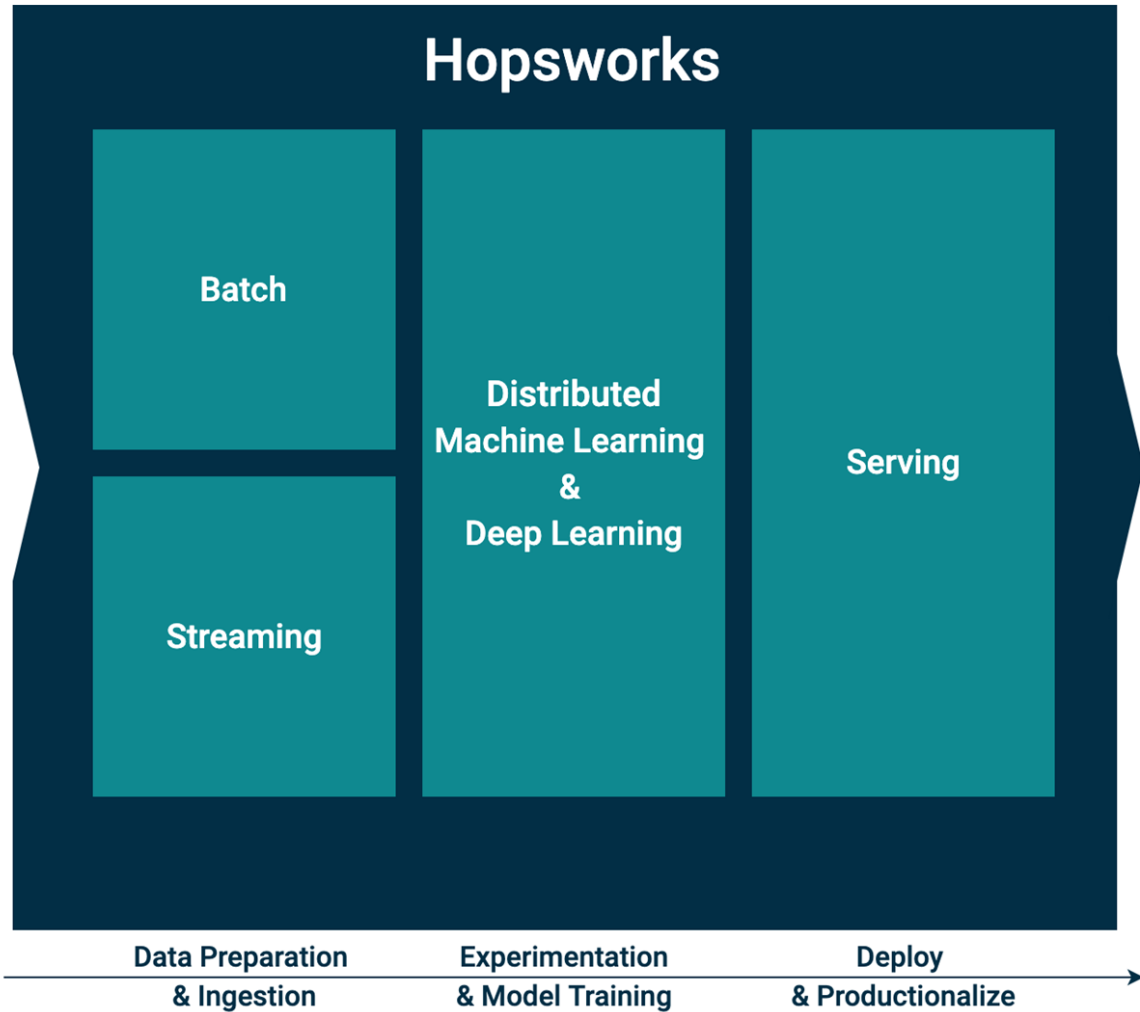
API

Dashboards

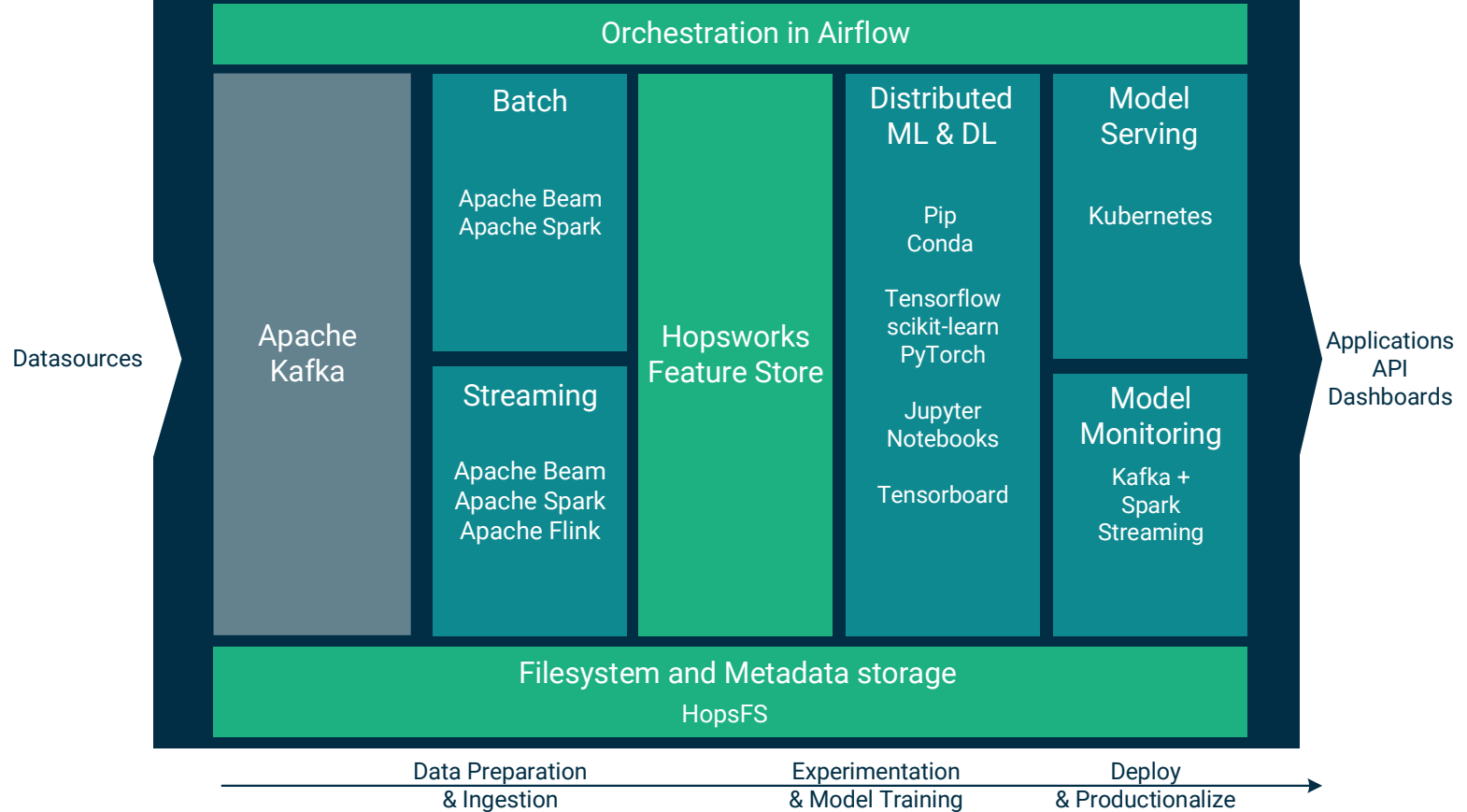
Data Preparation
& Ingestion

Experimentation
& Model Training

Deploy
& Productionalize

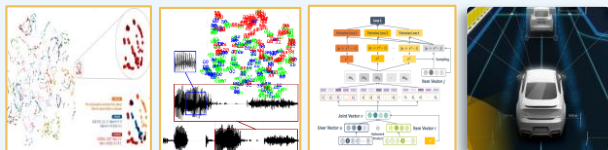


Hopsworks



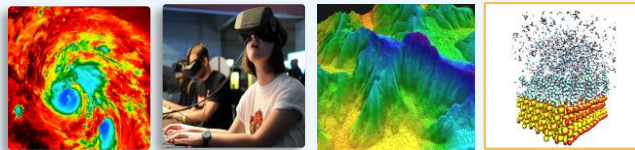


Open Source Foundation for Machine learning



TRANSLATE SPEECH RECOMMENDER SIMULATIONS

Machine Learning



WEATHER EDUCATION SEISMIC PROCESSING MOLECULAR SIMULATIONS

Scientific Apps

Open source stack

Latest ML Frameworks

Optimized Compilers, Math & Communication Libraries



Write parallel GPU algorithms entirely from Python™



Cluster Scale: Docker™, SLURM and Kubernetes Extensions

Frameworks



Middleware and Libraries

MIOpen

rocBLAS

rocFFT

rocPRIM

RCCL

Eigen

Runtimes & Drivers

OpenMP

HIP

OpenCL™

Python™

ROCm Kernel Driver

Devices

GPU

CPU

APU

DL Accelerator



Distro: Upstream Linux Kernel Support

ubuntu 



redhat.



CentOS



1100+ upstream ROCm driver commits since 4.12 kernel
<https://github.com/RadeonOpenCompute/ROCK-Kernel-Driver>

Languages: Multiple Programming options



LLVM

Programming Models

HIP

OpenMP

Python™

OpenCL™

LLVM -> AMDGCN Compiler

AMDGPU Code

LLVM: <https://llvm.org/docs/AMDGPUUsage.html>

CLANG HIP: https://clang.llvm.org/doxygen/HIP_8h_source.html



Machine Learning Frameworks

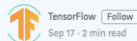


TensorFlow

PyTorch

ROCm enablement upstreamed to mainline and latest version supported
Support for major applications and benchmarks
Support for fp32 and fp16 precisions

Community supported AMD ROCm build for TensorFlow



TensorFlow Follow
Sep 17 · 2 min read

A guest post by Mayank Daga, Director, Deep Learning Software, AMD

Community Supported Builds

Build Type	Status	Artifacts
Linux AMD ROCm GPU Nightly	build passing	Nightly
Linux AMD ROCm GPU Stable Release	build running	Release

<https://medium.com/tensorflow/community-supported-amd-rocml-build-for-tensorflow-e8e9ac258369>

Available today as a Docker™ container or as Python™ PIP wheel

Support for automatic mixed precision via rocPyX (ROCm PyTorch Extensions)

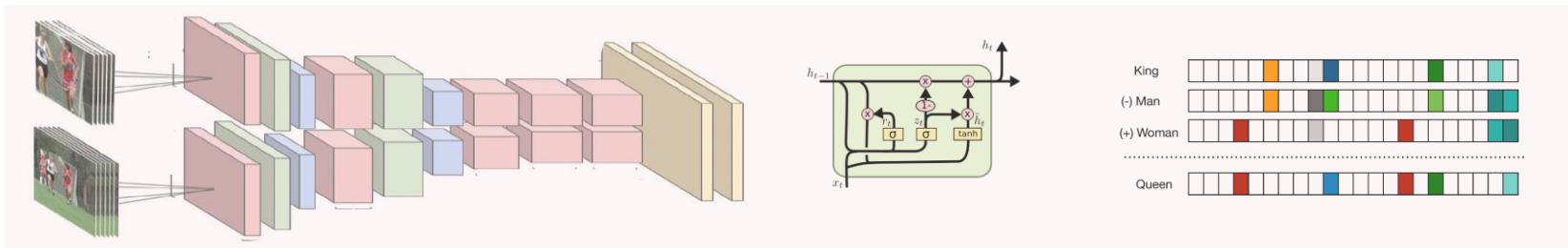
rocPyX

Available today as a Docker™ container or build from source

MIOpen: Open Source Machine Learning Library

Helps realize the incredible benefits of high-performance, highly-tuned Deep Learning primitives

- Single Precision (FP32), Half Precision (FP16), Mixed Precision and bFloat16 supported
- >500 Operators Accelerated. Hand tuned Assembly Operations



ACTIVATION ReLU Sigmoid TANH Leaky RELU	CONVOLUTION Winograd FFT Direct GEMM Implicit GEMM Dilated Convolutions Group Convolutions	POOLING Max, Average	NORMALIZATION LRN Batch Normalization	RNNs LSTMs GRU	Embeddings
Dropout SoftMax	Transpose Convolutions				



Support for Robust AI Compiler Technologies



XLA enabled on ROCm and Upstreamed
Performance improvements realized
over classic backend



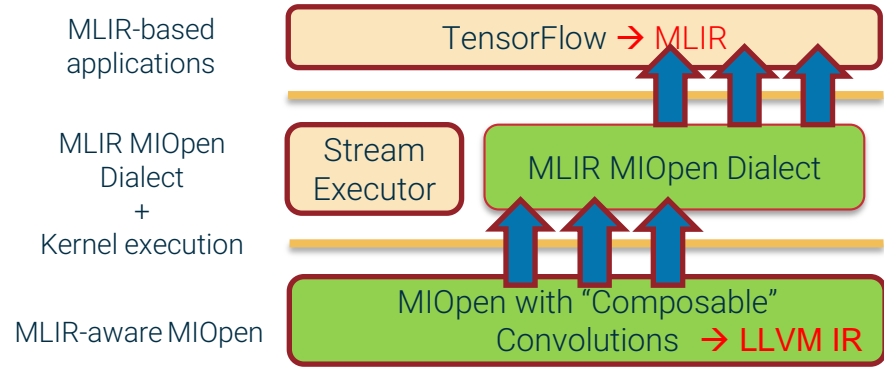
Basic functionality enabled on ROCm
Upstreaming in progress



Learn more about MLIR on
ROCm at [TensorFlow World](#)
in 2 weeks



AMD is a launch partner of Google's MLIR technology

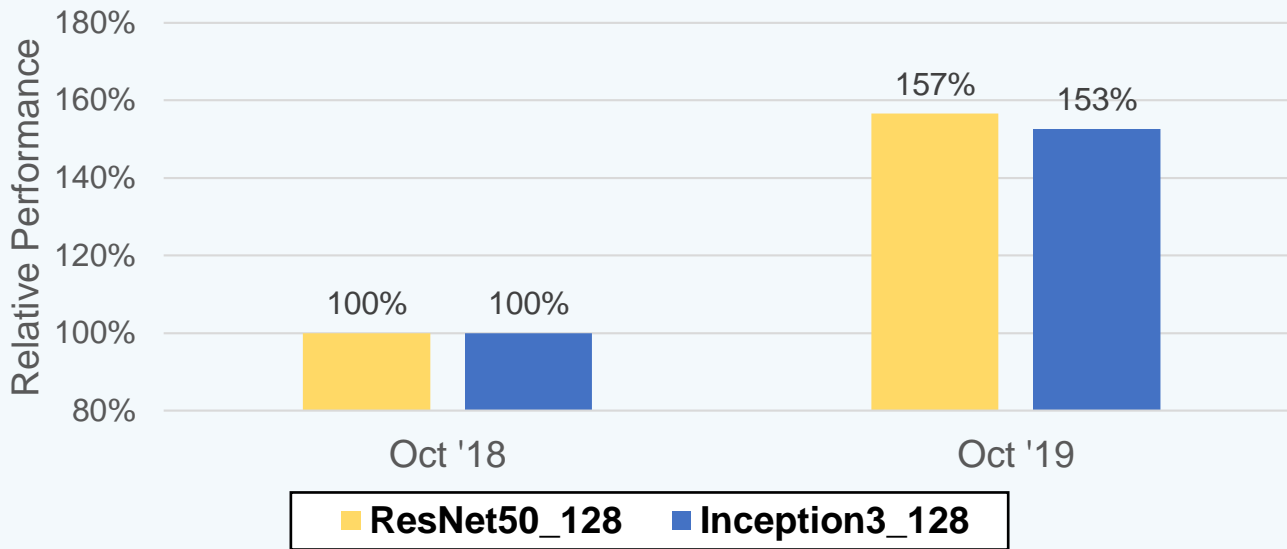


Eliminate "black-box" nature of Convolutions to drive next-gen performance optimizations



ROCm Performance Improvements

>1.5X Year-over-Year Software Performance Improvements on the Same Hardware



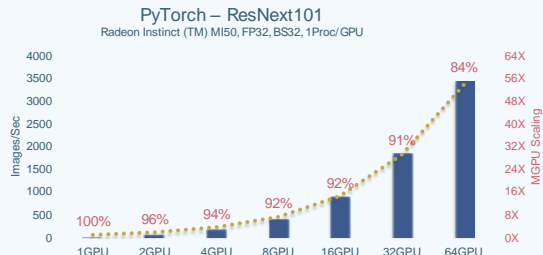
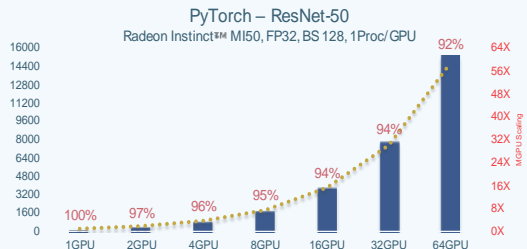
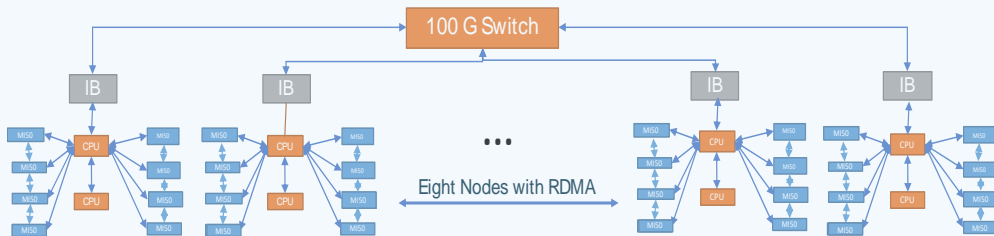
ROCm is on a monthly release cadence packed with new features and performance optimizations released both as source and docker containers

All performance measured on a single MI25 GPU using fp32

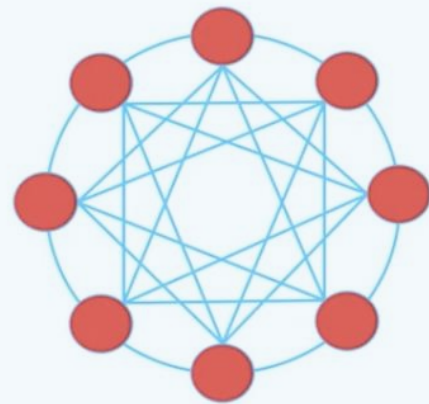


Intra-Node and Inter-Node ready

RCCL - Optimized collective communication operations library. Support for Infiniband and RoCE highspeed network fabrics. Designed for Easy MPI integration



OPEN
Compute Project®



Chordal Ring with Infinity Fabric ~ All to All Communication



Tools: System Management

ROCm Validation Suite (RVS)

A cluster management tool for detecting and troubleshooting software and hardware configuration issues, basic diagnostics, integration issues and system performance.

GPUP	GPU Properties
GM Module	GPU Monitoring
PESM Module	PCI Express State Monitor
RCQT Module	ROCm Configuration Qual Tool
PEQT Module	PCI Express Qualification Tool
SMQT Module	SBIOS Mapping Qualification Tool
PBQT	P2P Benchmark & Qualification Tool
PEBB Module	PCI Express Bandwidth Benchmark
GST Module	GPU Stress Test

ROCm System Management Interface (SMI)

A system administrator's tool for management and monitoring of GPU devices in ROCm enabled system.

```
--help --showID Show GPU ID
--showVer --showVer Show ROCm version
--showHW --showHW Show Hardware details
--showTemp --showTemp Show current temperature
--showClocks --showClocks Show current clock frequencies
--showFan --showFan Show current fan speed
--showSPM --showSPM Show current SPM Performance level
--showAvg --showAvg Show current Average Graphics Package Power Consumption
--showGPU --showGPU Show current GPU Clock OverDrive level
--showMem --showMem Show current GPU Memory Clock OverDrive level
--showPower --showPower Show maximum graphics package power this GPU will consume
--showProfile --showProfile Show Compute Profile attributes
--showClock --showClock Show supported GPU and Memory Clock
--showVer --showVer Show current GPU ver
--showMem --showMem Show current GPU memory used
--showPCI --showPCI Show estimated PCIe use
--showPCI --showPCI Show PCIe Replay Count
--showGPU --showGPU Show supported GPU and Memory Clocks and Voltages
--showVol --showVol Show current GPU voltage
--showInfo --showInfo Show relevant information and error results for the GPU

--resetProfile Reset Power Profile back to default.
--enable BLOCK ERRTYPE Enable RAS for specified block and error type
--disable BLOCK ERRTYPE Disable RAS for specified block and error type
--inject BLOCK Inject RAS poison for specified block (ONLY WORKS ON HAWAII/BOREAS)
--gpureset Reset specified GPU (One GPU must be specified)

--autoreply RESPONSE Response to automatically provide for all prompts (NOT RECOMMENDED)

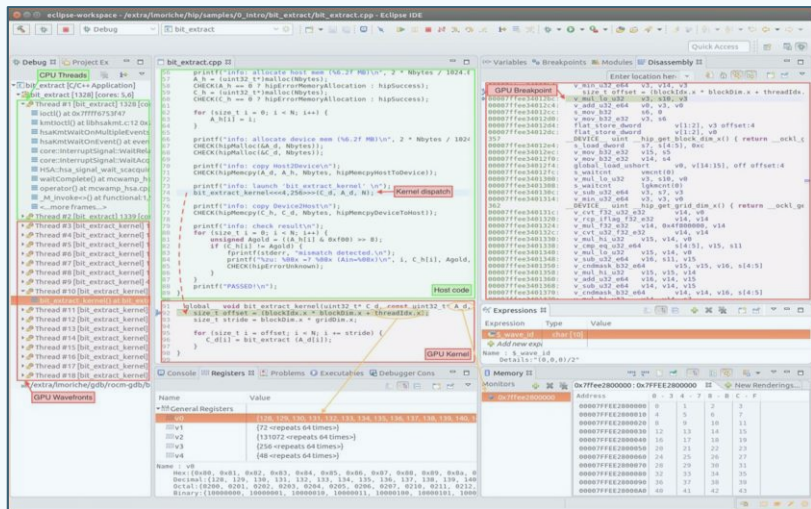
--logLevel LEVEL How much output will be printed for what program is doing, one of debug/info/warning/error/critical
--json Print output in JSON format
```




Tools: Debugger and Performance Profiling

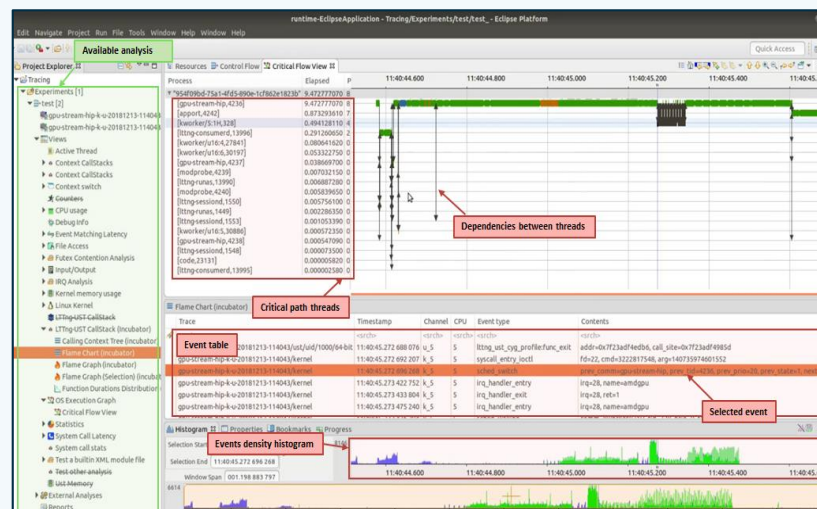
ROCM Debugger

A multi-architecture x86_64 CPU and amdgcN GPU debugger. Support for amdgcN ISA level and x86_64 debugging via GDB. There will be rocm debugger APIs available for 3rd party debugger developers.



ROCM Performance Profiling

rocProfiler allows collection of GPU HW Counter. rocTracer is a ondemand tracing with generic runtime's Events, Callback and Activity API. rocTX provides code annotation markers to profile specific sections of code.





ROCm: Machine Learning Applications

Image Classification

Object Detection

Machine Translation

Recommendation Systems

ResNet50, ResNet101
VGG
ResNext101
Inception3, Inception4
ShuffleNet, DenseNet
MobileNet, SqueezeNet
...

Faster-RCNN-Resnet50
Mask-RCNN-Resnet50
SSD-Resnet50
...

GNMT: LSTMs
Translate: LSTMs
BERT: Transformer
GPT-2: Transformer

DLRM
NCF

All state-of-the-art models are enabled on ROCm using TensorFlow & PyTorch

ResNext101 (fp32)
Batch=32
63 img/sec

Mask-RCNN-Resnet101(fp32)
Batch=2, Img:1Kx1K
144 img/sec

BERT
66 examples/sec

Neural Collaborative Filtering
2,961,459 tokens/sec

All performance measured on a single MI50 GPU using fp32



THE FIRST COMPLETE

OPEN SOFTWARE PLATFORM FOR COMPUTE

Call to Action: Download latest
ROCm release and contribute

Optimized for HPC and
Deep Learning at Scale

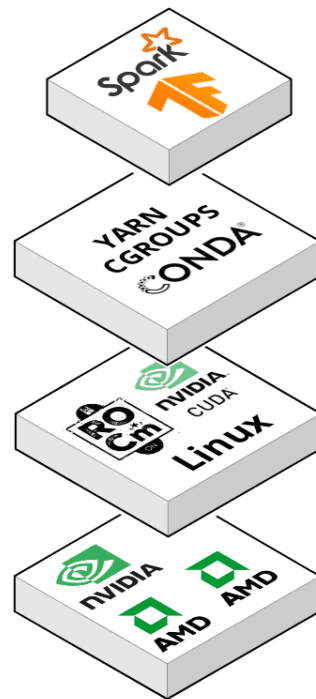
Enabling Innovation,
Collaboration, and Efficiency

ROCm in Hopsworks ML Pipelines

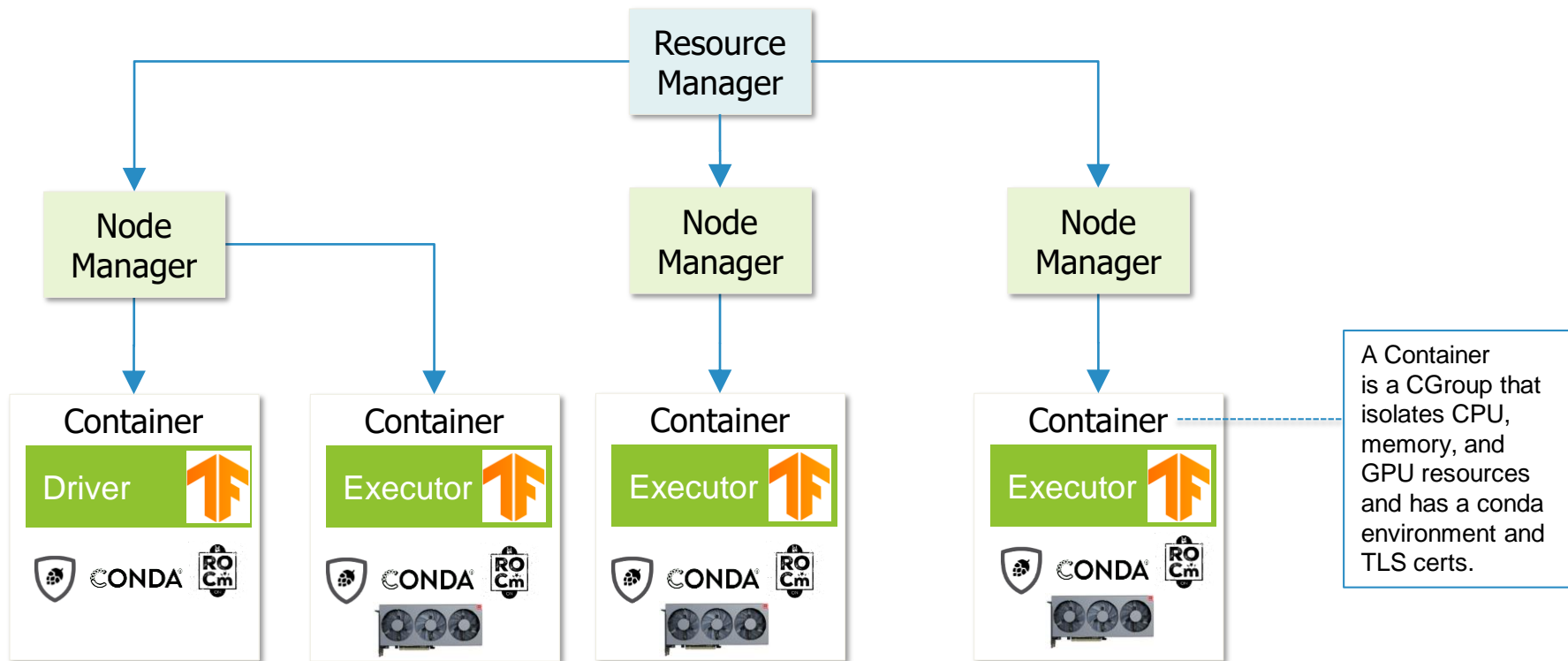
Spark / TensorFlow Applications on ROCm

Goal: Spark / TensorFlow applications in Hopsworks should run unchanged on ROCm

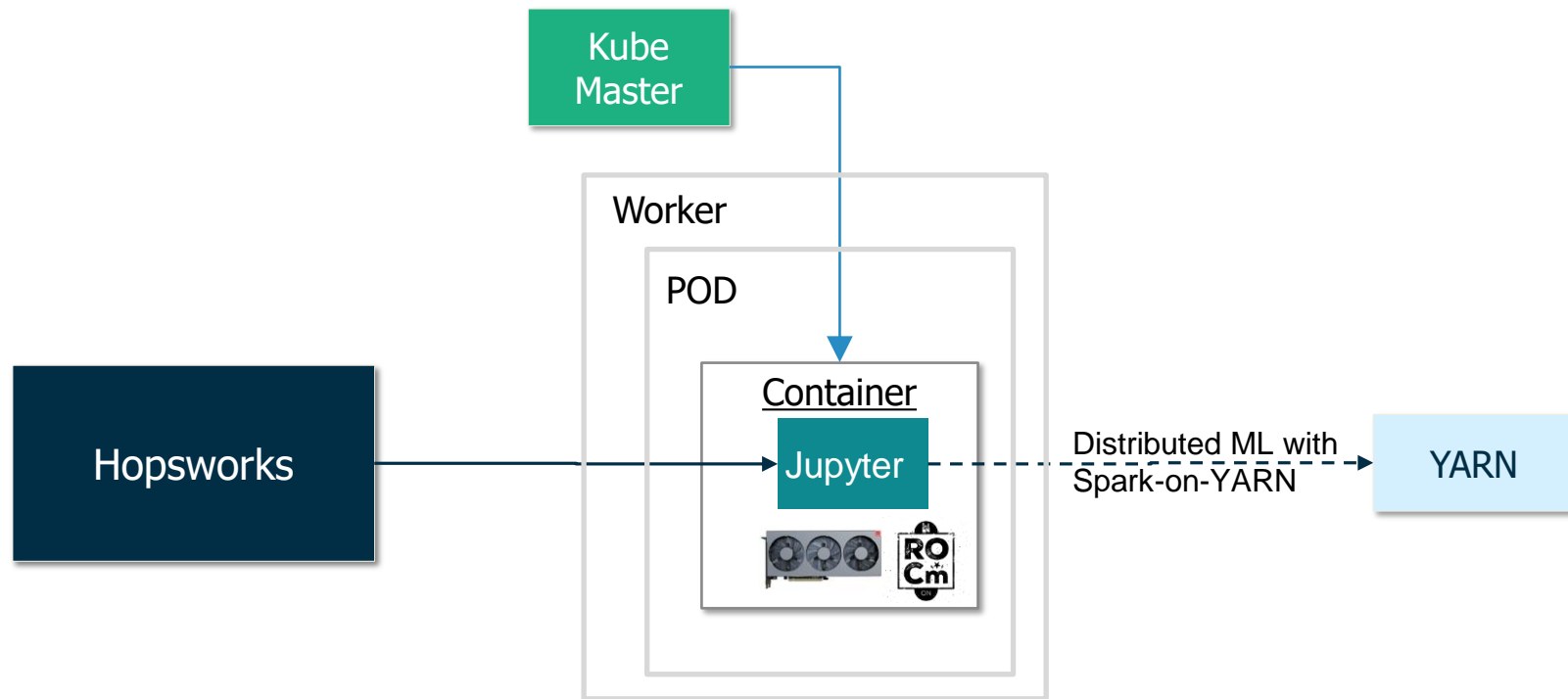
Solution: Hopsworks runs Spark/TensorFlow on YARN with support for ROCm through CGroups



YARN support for ROCm in Hopsworks

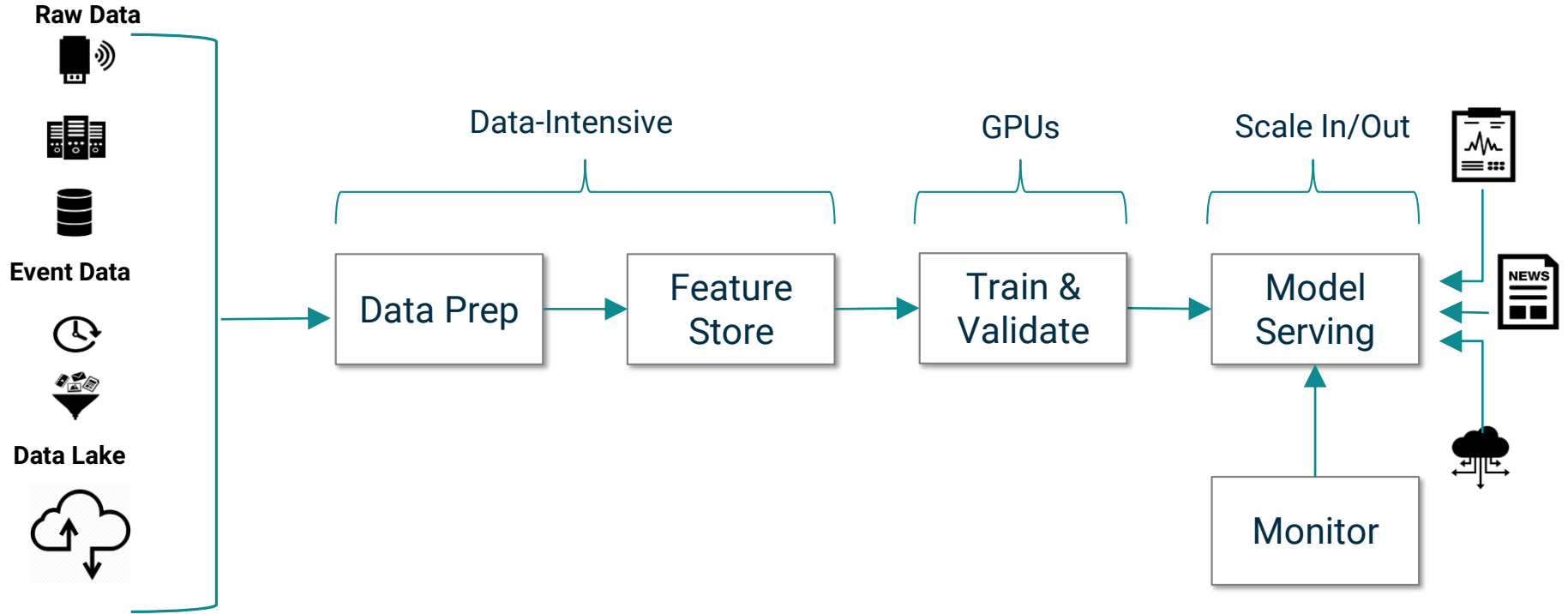


Kubernetes support for Jupyter/ROCM in Hopsworks

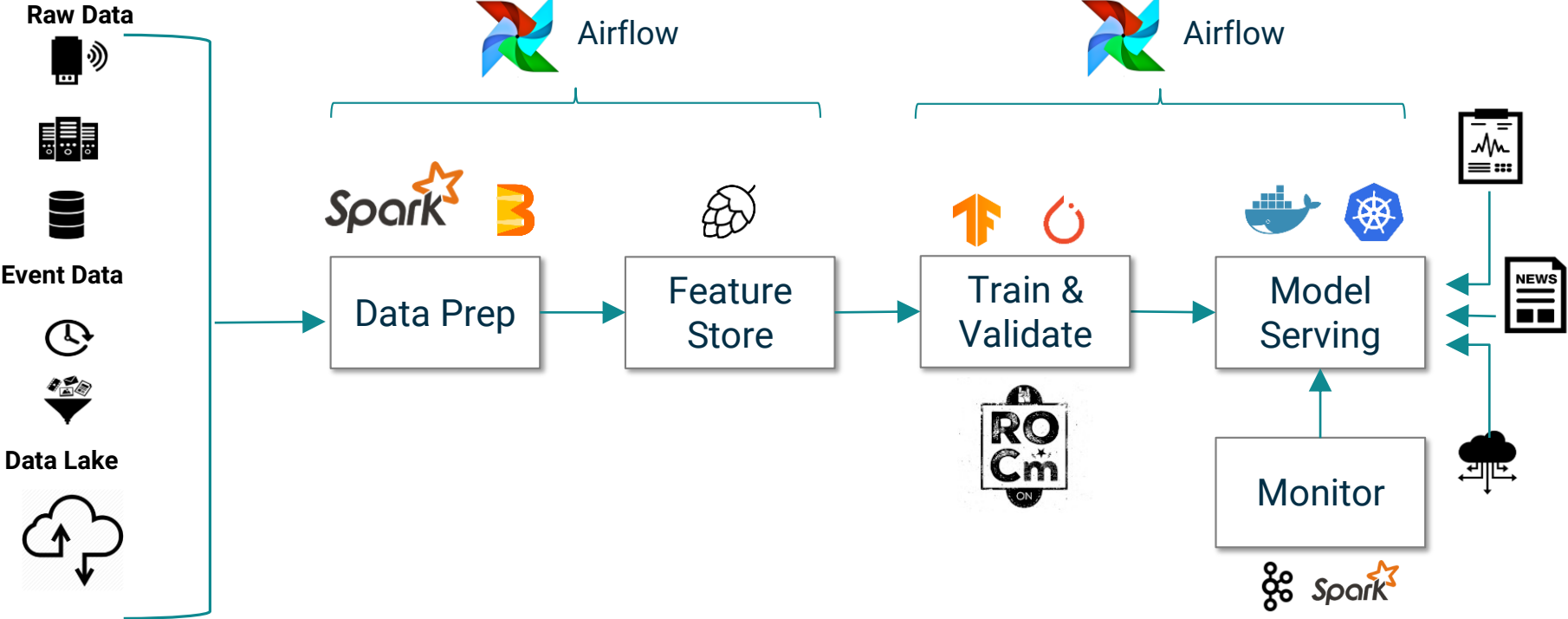


<https://kubernetes.io/docs/tasks/manage-gpus/scheduling-gpus/>

End-to-End ML Pipelines

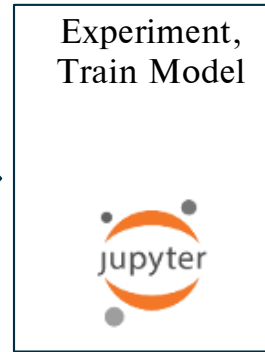
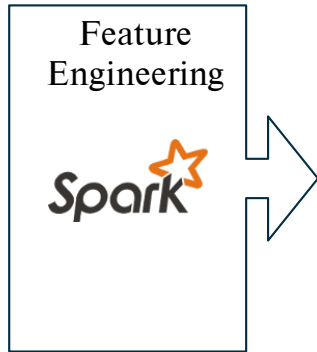


End-to-End ML Pipeline Technologies in Hopsworks



ML Pipelines of Jupyter Notebooks with Airflow

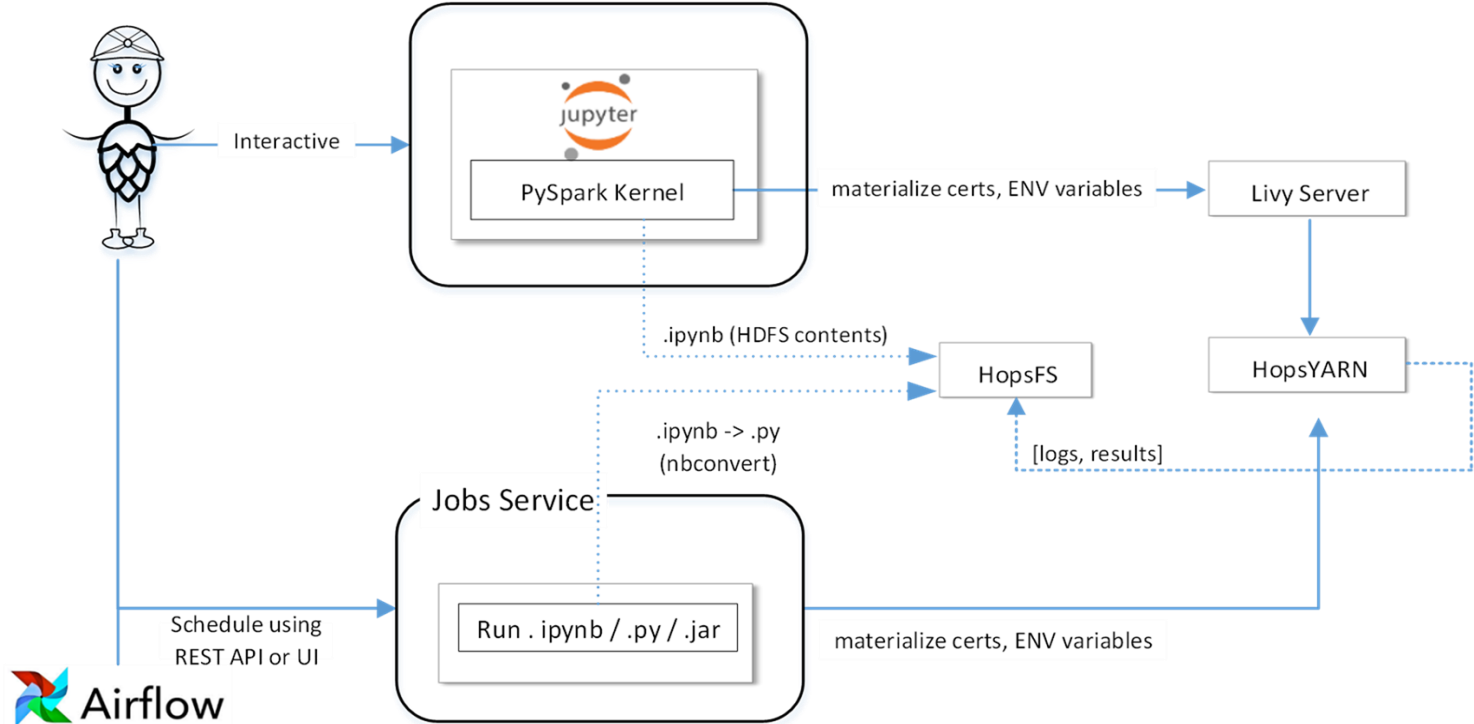
Dataprep Pipeline



Training and Deployment Pipeline



PySpark Notebooks as Jobs in ML Pipelines



Apache Airflow to Orchestrate ML Pipelines

The screenshot displays the Apache Airflow web interface for a DAG named 'txf_beam_pipeline'. The interface includes a navigation bar at the top with the Airflow logo and menu items: DAGs, Data Profiling, Browse, Admin, Docs, and About. The current date and time are 2019-06-17 20:58:00 UTC. Below the navigation bar, the DAG name 'txf_beam_pipeline' is shown with a 'off' status indicator and a 'schedule: *!77 * * * *' button. A toolbar contains various view options: Graph View (selected), Tree View, Task Duration, Task Tries, Landing Times, Gantt, Details, Code, Refresh, and Delete. Below the toolbar, there are filters for 'success' status, 'Base date: 2019-06-17 12:41:23', 'Number of runs: 25', 'Run: manual__2019-06-17T12:41:22.390484+00:00', and 'Layout: Left->Right'. A search bar is also present. The DAG is identified as 'HopworksLaunchOperator' and has a status legend showing 'success', 'running', 'failed', 'skipped', 'rescheduled', 'retry', 'queued', and 'no status'. The DAG graph consists of eight tasks: 'compute_statistics', 'infer_schema', 'compare_statistics', 'freeze_schema', 'preprocess_inputs', 'compute_stats_transformed_data', 'training', and 'tfma', all connected in a linear sequence.

Airflow DAG: txf_beam_pipeline schedule: *!77 * * * *

Graph View | Tree View | Task Duration | Task Tries | Landing Times | Gantt | Details | Code | Refresh | Delete

success Base date: 2019-06-17 12:41:23 Number of runs: 25 Run: manual__2019-06-17T12:41:22.390484+00:00 Layout: Left->Right Go Search for...

HopworksLaunchOperator success running failed skipped rescheduled retry queued no status

```
graph LR; compute_statistics --> infer_schema; infer_schema --> compare_statistics; compare_statistics --> freeze_schema; freeze_schema --> preprocess_inputs; preprocess_inputs --> compute_stats_transformed_data; compute_stats_transformed_data --> training; training --> tfma;
```

Apache Airflow to Orchestrate ML Pipelines



Jobs REST API

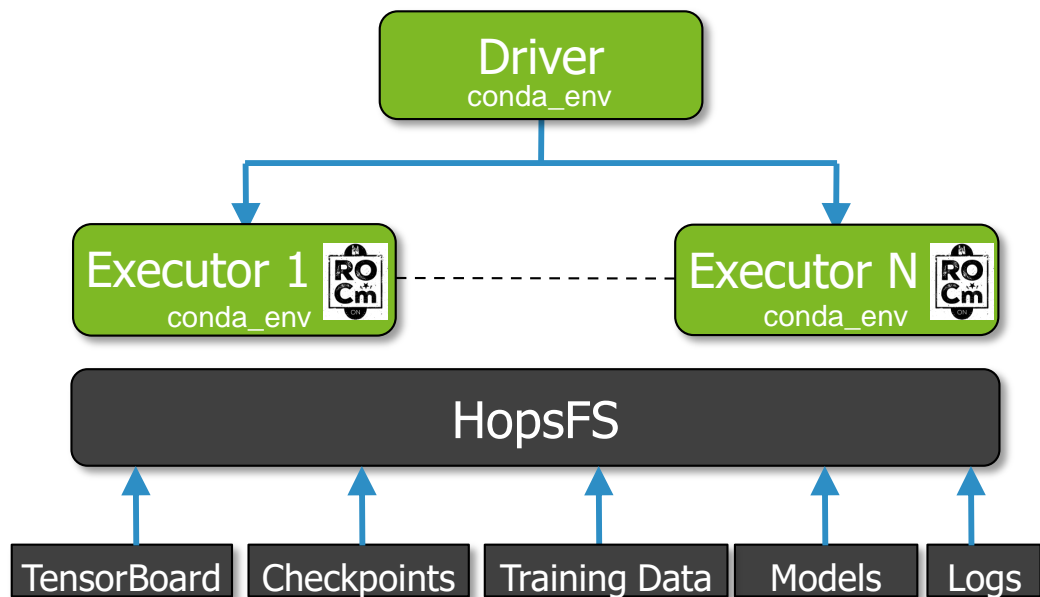
Hopsworks Jobs:
PySpark, Spark,
Flink, Beam/Flink

The screenshot shows the Hopsworks Jobs interface. The table contains the following data:

Name	Created on	Type	Owner	Submitted at	State	Status	Progress	Duration	Actions
TFMA	Jul 24, 2019 10:21:29 PM	PySpark	Theo Kakantousis	Jul 25, 2019 1:38:21 PM	Finished	Succeeded	100%	01:00	[Stop] [Refresh] [More]
Training	Jul 24, 2019 10:23:36 PM	PySpark	Theo Kakantousis	Jul 25, 2019 1:07:33 PM	Finished	Succeeded	100%	01:42	[Stop] [Refresh] [More]
ComputeStatsTransformedData	Jul 24, 2019 10:20:49 PM	PySpark	Theo Kakantousis	Jul 25, 2019 12:16:11 PM	Finished	Succeeded	100%	01:00	[Stop] [Refresh] [More]
PreProcessInputs	Jul 24, 2019 10:20:11 PM	PySpark	Theo Kakantousis	Jul 25, 2019 12:13:59 PM	Finished	Succeeded	100%	01:59	[Stop] [Refresh] [More]
FreezeSchema	Jul 24, 2019 10:19:40 PM	PySpark	Theo Kakantousis	Jul 25, 2019 12:13:15 PM	Finished	Succeeded	100%	00:23	[Stop] [Refresh] [More]
CompareStatistics	Jul 24, 2019 10:19:15 PM	PySpark	Theo Kakantousis	Jul 25, 2019 12:11:57 PM	Finished	Succeeded	100%	00:54	[Stop] [Refresh] [More]
InferSchema	Jul 24, 2019 10:18:38 PM	PySpark	Theo Kakantousis	Jul 25, 2019 3:05:56 PM	Running	Undefined	15%	00:10	[Stop] [Refresh] [More]
ComputeStatistics	Jul 24, 2019 10:18:00 PM	PySpark	Theo Kakantousis	Jul 25, 2019 3:04:38 PM	Finished	Succeeded	100%	01:01	[Stop] [Refresh] [More]
session	Jul 24, 2019 9:53:57 PM	Beam(Flink)	Theo Kakantousis	Jul 25, 2019 2:59:02 PM	Running	Undefined	100%	07:03	[Stop] [Refresh] [More]

Distributed Deep Learning with ROCm in Hopsworks

HParam Tuning with Spark+TensorFlow

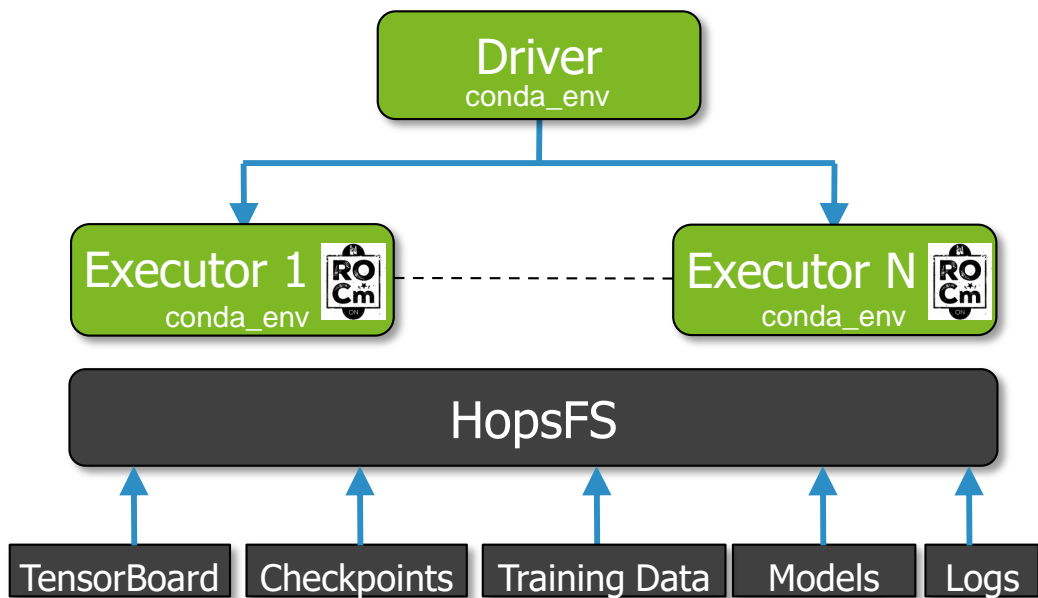


```
# RUNS ON THE EXECUTORS
def train(lr, dropout):
    def input_fn(): # return dataset
        optimizer = ...
        model = ...
        model.add(Conv2D(...))
        model.compile(...)
        model.fit(...)
        model.evaluate(...)

# RUNS ON THE DRIVER
Hparams= {'lr':[0.001, 0.0001],
         'dropout':[0.25, 0.5, 0.75]}
experiment.grid_search(train,Hparams)
```

<https://github.com/logicalclocks/hops-examples>

Distributed Training with Spark+TensorFlow



```
# RUNS ON THE EXECUTORS
def train():
    def input_fn(): # return dataset
        model = ...
        optimizer = ...
        model.compile(...)
    rc = tf.estimator.RunConfig(
        'CollectiveAllReduceStrategy')
    keras_estimator = tf.keras.estimator.
        model_to_estimator(...)
    tf.estimator.train_and_evaluate(
        keras_estimator, input_fn)
# RUNS ON THE DRIVER
experiment.collective_all_reduce(train)
```

<https://github.com/logicalclocks/hops-examples>

Demo



DEMO

End-to-End ML Workflow with Spark/TensorFlow and ROCm

That was Hopsworks with ROCm



Efficiency & Performance



Feature Store

Data warehouse for ML



Distributed Deep Learning

Faster with more GPUs



HopsFS

NVMe speed with Big Data



Horizontally Scalable

Ingestion, DataPrep,
Training, Serving

Development & Operations



Development Environment

First-class Python Support



Version Everything

Code, Infrastructure, Data



Model Serving on Kubernetes

TF Serving, SkLearn



End-to-End ML Pipelines

Orchestrated by Airflow

Security & Governance



Secure Multi-Tenancy

Project-based restricted access



Encryption At-Rest, In-Motion

TLS/SSL everywhere



AI-Asset Governance

Models, experiments, data, GPUs



Data/Model/Feature Lineage

Discover/track dependencies

LOGICAL
CLOCKS

@logicalclocks 

www.logicalclocks.com

AMD 



OPEN SOURCE
FOUNDATION
FOR MACHINE
LEARNING

Acknowledgements and References

Slides and Diagrams from colleagues:

- Maggy: Moritz Meister and Sina Sheikholeslami
- Feature Store: Kim Hammar
- Beam/Flink on Hopsworks: Theofilos Kakantousis

References

- HopsFS: Scaling hierarchical file system metadata ..., USENIX FAST 2017.
- Size matters: Improving the performance of small files ..., ACM Middleware 2018.
- ePipe: Near Real-Time Polyglot Persistence of HopsFS Metadata, CCGrid, 2019.
- Hopsworks Demo, SysML 2019.

Disclaimer & Attribution

The information presented in this document is for informational purposes only and may contain technical inaccuracies, omissions and typographical errors.

The information contained herein is subject to change and may be rendered inaccurate for many reasons, including but not limited to product and roadmap changes, component and motherboard version changes, new model and/or product releases, product differences between differing manufacturers, software changes, BIOS flashes, firmware upgrades, or the like. AMD assumes no obligation to update or otherwise correct or revise this information. However, AMD reserves the right to revise this information and to make changes from time to time to the content hereof without obligation of AMD to notify any person of such revisions or changes.

AMD MAKES NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE CONTENTS HEREOF AND ASSUMES NO RESPONSIBILITY FOR ANY INACCURACIES, ERRORS OR OMISSIONS THAT MAY APPEAR IN THIS INFORMATION.

AMD SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE. IN NO EVENT WILL AMD BE LIABLE TO ANY PERSON FOR ANY DIRECT, INDIRECT, SPECIAL OR OTHER CONSEQUENTIAL DAMAGES ARISING FROM THE USE OF ANY INFORMATION CONTAINED HEREIN, EVEN IF AMD IS EXPRESSLY ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

ATTRIBUTION

© 2019 Advanced Micro Devices, Inc. All rights reserved. AMD, the AMD Arrow logo, Radeon and combinations thereof are trademarks of Advanced Micro Devices, Inc. in the United States and/or other jurisdictions. Other names are for informational purposes only and may be trademarks of their respective owners. Use of third party marks / names is for informational purposes only and no endorsement of or by AMD is intended or implied.