ROCM 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.

Monoplies are not good for deep learning!
## Nvidia™ 2080Ti vs AMD Radeon™ VII: ResNet-50

<table>
<thead>
<tr>
<th>Nvidia™ 2080Ti</th>
<th>AMD Radeon™ VII</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Memory:</strong> 11GB</td>
<td><strong>Memory:</strong> 16 GB</td>
</tr>
<tr>
<td>TensorFlow 1.12</td>
<td>TensorFlow 1.14</td>
</tr>
<tr>
<td>CUDA 10.0.130, cuDNN 7.4.1</td>
<td>ROCm: 2.7</td>
</tr>
<tr>
<td><strong>Model:</strong> RESNET-50</td>
<td><strong>Model:</strong> RESNET-50</td>
</tr>
<tr>
<td><strong>Dataset:</strong> imagenet (synthetic)</td>
<td><strong>Dataset:</strong> imagenet (synthetic)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>FP32 total images/sec:</th>
<th>FP16 total images/sec:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nvidia™ 2080Ti</td>
<td>~322</td>
<td>~560</td>
</tr>
<tr>
<td>AMD Radeon™ VII</td>
<td>~316</td>
<td>~421</td>
</tr>
</tbody>
</table>

References:
- [https://lambdalabs.com/blog/2080-ti-deep-learning-benchmarks/](https://lambdalabs.com/blog/2080-ti-deep-learning-benchmarks/)
- [https://github.com/ROCmSoftwarePlatform/tensorflow-upstream/issues/173](https://github.com/ROCmSoftwarePlatform/tensorflow-upstream/issues/173)
Hopsworks hides the Complexity of Deep Learning

Hopsworks Feature Store

[Adapted from Schulley et al. “Technical Debt of ML”]
Hopsworlks

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

### Machine Learning

<table>
<thead>
<tr>
<th>Frameworks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Caffe2</td>
</tr>
<tr>
<td>PyTorch</td>
</tr>
<tr>
<td>TensorFlow</td>
</tr>
<tr>
<td>ONNX</td>
</tr>
</tbody>
</table>

### Scientific Apps

<table>
<thead>
<tr>
<th>Open source stack</th>
</tr>
</thead>
<tbody>
<tr>
<td>Latest ML Frameworks</td>
</tr>
<tr>
<td>Optimized Compilers, Math &amp; Communication Libraries</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Middleware and Libraries</th>
</tr>
</thead>
<tbody>
<tr>
<td>MIOpen</td>
</tr>
<tr>
<td>rocBLAS</td>
</tr>
<tr>
<td>rocFFT</td>
</tr>
<tr>
<td>rocPRIM</td>
</tr>
<tr>
<td>RCCL</td>
</tr>
<tr>
<td>Eigen</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Runtimes &amp; Drivers</th>
</tr>
</thead>
<tbody>
<tr>
<td>OpenMP</td>
</tr>
<tr>
<td>HIP</td>
</tr>
<tr>
<td>OpenCL™</td>
</tr>
<tr>
<td>Python™</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Devices</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPU</td>
</tr>
<tr>
<td>CPU</td>
</tr>
<tr>
<td>APU</td>
</tr>
<tr>
<td>DL Accelerator</td>
</tr>
</tbody>
</table>
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**

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

Available today as a Docker™ container or build from source

**PyTorch**

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

Available today as a Docker™ container or build from source

---

Community supported AMD ROCm build for TensorFlow

<table>
<thead>
<tr>
<th>Build Type</th>
<th>Status</th>
<th>Artifacts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linux AMD ROCm GPU Nightly</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Linux AMD ROCm GPU Stable Release</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

https://medium.com/tensorflow/community-supported-amd-rocm-build-for-tensorflow-e8e9ac258369

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

<table>
<thead>
<tr>
<th>ACTIVATION</th>
<th>CONVOLUTION</th>
<th>POOLING</th>
<th>NORMALIZATION</th>
<th>RNNs</th>
<th>Embeddings</th>
</tr>
</thead>
<tbody>
<tr>
<td>ReLU</td>
<td>Winograd</td>
<td>Max, Average</td>
<td>LRN</td>
<td>LSTMs</td>
<td></td>
</tr>
<tr>
<td>Sigmoid</td>
<td>FFT</td>
<td></td>
<td>Batch Normalization</td>
<td>GRU</td>
<td></td>
</tr>
<tr>
<td>TANH</td>
<td>Direct GEMM</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Leaky RELU</td>
<td>Implicit GEMM</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dropout</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SoftMax</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Transpose Convolutions
- Dilated Convolutions
- Group Convolutions

https://github.com/ROCmSoftwarePlatform/MIOpen
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
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.
RCCL - Optimized collective communication operations library. Support for Infiniband and RoCE highspeed network fabrics. Designed for Easy MPI integration.
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.

<table>
<thead>
<tr>
<th>Module</th>
<th>Tool Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPU Properties</td>
<td></td>
</tr>
<tr>
<td>GM Module</td>
<td>GPU Monitoring</td>
</tr>
<tr>
<td>PESM Module</td>
<td>PCI Express State Monitor</td>
</tr>
<tr>
<td>RCQT Module</td>
<td>ROCm Configuration Qual Tool</td>
</tr>
<tr>
<td>PEQT Module</td>
<td>PCI Express Qualification Tool</td>
</tr>
<tr>
<td>SMQT Module</td>
<td>SBIOS Mapping Qualification Tool</td>
</tr>
<tr>
<td>PBQT</td>
<td>P2P Benchmark &amp; Qualification Tool</td>
</tr>
<tr>
<td>PEBB Module</td>
<td>PCI Express Bandwidth Benchmark</td>
</tr>
<tr>
<td>GST Module</td>
<td>GPU Stress Test</td>
</tr>
</tbody>
</table>

https://github.com/ROCm-Developer-Tools/ROCmValidationSuite

**ROCm System Management Interface (SMI)**
A system administrator’s tool for management and monitoring of GPU devices in ROCm enabled system.

https://github.com/RadeonOpenCompute/ROC-smi
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.

https://github.com/ROCm-Developer-Tools/rocprofiler
https://github.com/ROCm-Developer-Tools/roctracer
## ROCm: Machine Learning Applications

<table>
<thead>
<tr>
<th></th>
<th>Image Classification</th>
<th>Object Detection</th>
<th>Machine Translation</th>
<th>Recommendation Systems</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResNet50, ResNet101</td>
<td>VGG</td>
<td>Faster-RCNN-Resnet50</td>
<td>GNMT: LSTMs</td>
<td>DLRM</td>
</tr>
<tr>
<td>ResNext101</td>
<td>Inception3, Inception4</td>
<td>Mask-RCNN-Resnet50</td>
<td>Translate: LSTMs</td>
<td>NCF</td>
</tr>
<tr>
<td>VGG</td>
<td>ResNext101</td>
<td>SSD-Resnet50</td>
<td>BERT: Transformer</td>
<td></td>
</tr>
<tr>
<td>MobileNet, SqueezeNet</td>
<td>Inception3, Inception4</td>
<td>...</td>
<td>GPT-2: Transformer</td>
<td></td>
</tr>
<tr>
<td>ResNet50, ResNet101</td>
<td>Faster-RCNN-Resnet50</td>
<td>...</td>
<td>...</td>
<td></td>
</tr>
</tbody>
</table>

### Performance

<table>
<thead>
<tr>
<th>Model</th>
<th>Tensorflow &amp; PyTorch</th>
<th>TensorFlow</th>
<th>PyTorch</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResNext101 (fp32)</td>
<td>Batch=32, 63 img/sec</td>
<td>63 img/sec</td>
<td></td>
</tr>
<tr>
<td>Mask-RCNN-Resnet101 (fp32)</td>
<td>Batch=2, Img:1Kx1K</td>
<td>144 img/sec</td>
<td></td>
</tr>
<tr>
<td>BERT</td>
<td>66 examples/sec</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Neural Collaborative Filtering</td>
<td>2,961,459 tokens/sec</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

All performance measured on a single MI50 GPU using fp32.
NEW ERA OF GPU ACCELERATED MACHINE LEARNING

OPEN SOFTWARE PLATFORM FOR COMPUTE

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

A Container is a CGroup that isolates CPU, memory, and GPU resources and has a conda environment and TLS certs.
Kubernetes support for Jupyter/ROCm in Hopsworks

https://kubernetes.io/docs/tasks/manage-gpus/scheduling-gpus/
End-to-End ML Pipelines

Raw Data

Data Prep

Feature Store

Train & Validate

Model Serving

Scale In/Out

Event Data

Data Lake

Data-Intensive

GPUs

Monitor

Raw Data

Feature Store

Train & Validate

Model Serving

Scale In/Out

Event Data

Data Lake

Data-Intensive

GPUs

Monitor
End-to-End ML Pipeline Technologies in Hopsworks

Raw Data

Event Data

Data Lake

Data Prep

Feature Store

Train & Validate

Model Serving

Monitor

Airflow

Spark

Feature Store

Train & Validate

Model Serving

Monitor

Airflow

Airflow
ML Pipelines of Jupyter Notebooks with Airflow

Dataprep Pipeline

Feature Engineering

Feature Store

Training and Deployment Pipeline

Select Features, File Format

Experiment, Train Model

Validate & Deploy Model
Don’t just Throw Away Notebooks

Notebook → port/rewrite code → Python + YML in Git

Explore Data, Train model → Hyperparam Opt. → Distributed Training

Iteration is hard/impossible/a-bad-idea
PySpark Notebooks as Jobs in ML Pipelines
Apache Airflow to Orchestrate ML Pipelines
Apache Airflow to Orchestrate ML Pipelines

Airflow

Jobs REST API

Hopsworks Jobs: PySpark, Spark, Flink, Beam/Flink
Distributed Deep Learning with ROCm in Hopsworks
HParam Tuning with Spark+TensorFlow

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

# RUNS ON THE DRIVER
```python
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

```python
# 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
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
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
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.