Distributed Deep Learning with Containers on Heterogeneous GPU clusters

Dong Meng - dmeng@mapr.com
Data Scientist, MapR Technologies
Roadmaps

• Enterprise Big data Journey
• Distributed Deep Learning
• Apply Container Technology and NVIDIA GPUs
• Demo: Face recognition in video streams
Enterprise Big data Journey
Great buildings have great foundations
MapR Converged Data Platform

Shared Services

Files, Tables, Streams
together on same platform

Cloud-scale Data Store
Analytics & Machine Learning Engines
Operational Database
Event Data Streams

Converge-X™ Data Fabric

High Availability  Real Time  Security & Governance  Multi-tenancy  Disaster Recovery  Global Namespace

On-Premise, In the Cloud, Hybrid

Cloud-scale Data Store
Analytics & Machine Learning Engines
Operational Database
Event Data Streams

Converge-X™ Data Fabric

High Availability  Real Time  Security & Governance  Multi-tenancy  Disaster Recovery  Global Namespace

On-Premise, In the Cloud, Hybrid
An Enterprise Foundation To Operationalize Data

Existing Enterprise Applications

POSIX, NFS
HDFS API
HBase API
JSON API
Kafka API

Cloud-scale Data Store
Analytics & Machine Learning Engines
Operational Database
Event Data Streams

Converge-X™ Data Fabric

High Availability  Real Time  Security & Governance  Multi-tenancy  Disaster Recovery  Global Namespace

On-Premise, In the Cloud, Hybrid

Supports Open-Source APIs with Patented Speed, Scale, Reliability

Data Center

MapR Confidential © 2017 MapR Technologies
From a User Perspective

```
$ ls -F
apache-kylin-0.7.2-incubating-src-source-release.tar.gz
apache-kylin-0.7.2-incubating-src-source-release.sha1
bar/
build.xml
car-data.csv
coc.parquet/
counts.parquet/
deep.json
drillbit.log
dges.ssv
dges.csv.bak
```

```
$ pwd
/mapr/se1/user/tdunning
```

Directories

Streams

Files

Table
What we have observed:

• Enterprise customers start to gain great value from big data platform as technologies grow mature
• The data pipeline is growing from batch to real time streaming
• The volume of data collected is growing from TB to PB to EB.
• Customers are more motivated than ever to build intelligence applications based on collected data
• Machine learning use cases grow
• Add GPU cards to clusters
Distributed Deep Learning
More

DATA beats complex algorithms

The Unreasonable Effectiveness of Data, published by Google
Why Deep Learning

• Machine learning algorithm’s performance depend on the representation of the data they are given (feature engineering)
• Difficulty in finding the right representations (features)
• Deep learning
  – Learns multiple levels of representations
  – Learns high level of abstractions
• Growth of data volumes and computing power (GPUs)
Distributed Deep Learning System

From a system design perspective:
• Distributed Data Storage – MapRFS, HDFS, Ceph
• Consistency – Parameter Server
• Fault Tolerance – Checkpoint reload
• Resource management – Yarn, Mesos, Kubernetes
• Programming Mode – TensorFlow, Apache MXNet, Torch, Caffe
Distributed Deep Learning Model

Data Parallelism:
• Data Partition
• Train each model on mini-batches of data

Model Parallelism:
• Model Partition
• Separate the training task for each layer
Distributed Deep Learning System

From a user perspective:
- Ease of expression: for lots of crazy ML ideas/algorithms
- Scalability: can run experiments quickly
- Portability: can run on wide variety of platforms
- Reproducibility: easy to share and reproduce research
- Production readiness: go from research to real products
Apply Container Technology and NVIDIA GPUs
Deep Learning Development Environment

Individual Research:
• Laptop/Dev boxes

Research Lab Setting:
• HPC clusters, high speed job execution, small teams

Enterprise Deep learning System:
• Distributed Data Storage, Consistency, Fault Tolerance, Resource management, Programming Mode, Version and Deployment
• Container, Kubernetes, MapR Data Platform
Containers are Great for Deployment and Research

Advantages

- Reproducible work environments
- Ease of deployment
- Isolation of individual workspaces
- Run across heterogeneous environments
- Facilitate collaboration
Stateful Containers for Deep Learning

Advantages

- Containerized workspaces
  - Keep your work between sessions
- Manage work across many projects
- Work with versioned datasets and models
- Share work across containers, projects and/or teams
Microservices for Serving Deep Learning Models

Advantages

• Deploy models to production as microservices
• Use files, real-time streams and databases in production
• Scales horizontally
• Support both real-time and batch
• May or may not be stateful
Kubernetes in a Heterogeneous GPU Cluster

- Kubernetes master:
  - CPU-only
- Workers:
  - NVIDIA driver
  - CUDA
  - CUDNN
- Kubelet config updated for GPU workers needed
  ```bash
  --feature gates=Accelerators=true
  ```
- Note: Containers also need to be configured for GPU support separately

Diagram: Frederic Tausch on Github
MapR Volumes

/Projects
  /project1

/Users
  /jsmith
  /mjohnson

Data Nodes
Architecture

Application layer

Orchestration layer

Data layer

Parameter Server

TF Trainer 1

TF Trainer 2

Pod 1

Pod 2

Pod 3

kubernetes Master

Enterprise Storage
MapR-FS

NoSQL Database
MapR-DB (binary, JSON)

Event Streaming (Kafka)
MapR-ES
Kubernetes for Containers Orchestration

• Deploy once and run many times
• Rollout new versions/rollback to old versions
• Dynamic Scheduling and Elastic Scaling
• Auto Fault Recovery and Scalable Computing
• Isolation and Quota
• Manage GPU resources
• Mount MapR Volume to Persistent Volume
MapR as the Infrastructure for Distributed DL

- Converged Data Platform for both Big Data and Deep Learning
- By design Volume Topology, collocate the data with GPU computation/training.
- Performant POSIX file system, quick adapt to new deep learning technologies and frameworks
- Mirroring feature enables model training/deployment with global data centers
- Snapshot feature enables research reproducibility
- MapRDB and MapR Streams provides infrastructures for Intelligence applications with deep learning models
Pattern 1: GPU Server as MapR Client

Dockerized GPU-based NVIDIA Tier, NVIDIA DGX systems as MapR client (High Performance POSIX Client)
Pattern 2: Collocated MapR + Kubernetes

NVIDIA GPU Cards scales vertically, storage scales horizontally
Demo: Real Time Streams Face Recognition on Pub/Sub Systems

GLOBAL DATA PLANE

Camera to capture video feeds

MAPR EDGE
Deploy Model at the edge to score data feed

Aggregate, analyze data at core and refine models

Send updated model back to edge

MAPR CONVERGED ENTERPRISE EDITION

[on-prem, hybrid, cloud]

Small footprint at the edge

MAPR EDGE

Reliable replication

Convergence at the edge
Demo: Real Time Streams Face Recognition on Pub/Sub Systems

The task: to identify a person in a video stream with a photo of hers/his.
Demo: Real Time Streams Face Recognition on Pub/Sub Systems

Producers to capture video feeds

DL model process streams

Consumers to output processed videos
Q&A

ENGAGE WITH US

Dong Meng
dmeng@mapr.com