GPU-ACCELERATING A DEEP LEARNING ANOMALY DETECTION PLATFORM

Satish Dandu, Michael Balint, and Joshua Patterson

nVIDIA
AGENDA

- Data Platform-as-a-Service
- Multi-Tenancy
- Anomaly Detection Platform
  - Training & Inferencing Evolution
  - Performance & Learnings
- GPU Acceleration
  - Dash boarding use case & impetus for GPU-acceleration
  - Performance & Learnings
- Future & GOAI
DATA PAAS OVERVIEW
DATA PLATFORM-AS-A-SERVICE

**SCALE**
- Handles 1M events/second
- Auto-scales the cluster automatically

**HIGH AVAILABILITY**
- Offers HA with no data-loss
- Always-on architecture
- Data replication

**SECURITY**
- Data platform security has been implemented with VPCs in AWS
- Dashboard access using NVIDIA LDAP

**SELF SERVICE**
- Log-to-analytics
- Kibana, JDBC access
- Accessing data using BI tools
ARCHITECTURE

V1
DATA PLATFORM STATS

10.066
95th percentile of Value

60
95th percentile of SLA
ANOMALY DETECTION
DATA PAAS

Anomaly Detection (AD) PaaS

*Images created from quickmeme.com*
ANOMALY DETECTION USING DEEP LEARNING

Data Platform

AI Framework (Keras + TensorFlow)

GPU Cloud

Anomaly Detection

Top Features

Automated Alerts & Dashboards
Early Detection
Self Service
Better accuracy & less noise
ANOMALY DETECTION FRAMEWORK

Supervised Learning: Logistic Regression

Unsupervised Learning: Multivariate-Gaussian

Feature Learning Algorithm: Recurrent Neural Network (RNN), Autoencoders (AE)

Anomalies: Email alerts, Dashboards

Anomaly Post-processing: Univariate Analysis

Feedback from user

Raw Dataset

Time | X1 | X2 | Y | Anomaly Description
--- | --- | --- | --- | ---

1 | X1
0

Time | X1 | X2 | X’ | X” | Y
--- | --- | --- | --- | --- | ---

1
0

Time | X1 | X2 | X’ | X”
--- | --- | --- | --- | ---

Anomaly Detection Feedback from user

Anomaly Post-processing: Univariate Analysis

Time | X1 | X2 |
--- | --- | --- |

Anomaly Description 1 | X1
0

Time | X1 | X2 |
--- | --- | --- |
ANOMALY DETECTION BENEFITS WITH DEEP LEARNING

With DL

Top Features

- Automated Alerts & Dashboards
- Early Detection
- Self Service
- Better accuracy & less noise

Without DL
ANOMALY DETECTION TRAINING

- Evolution
- CPU vs GPU
- Learnings:
  - Manual feature extraction does not scale
  - Dataset preparation is the long pole
  - Training on CPU takes longer than data collection rate
ANOMALY DETECTION INFERENCING

1. Aggregated Data output for live data
2. Deep learning Model
3. AD platform uses trained model to inference anomalies
4. Sends automated alerts with dashboards
5. Users can label alerts
6. Model gets updated during next training cycle

Goal: Detect/Inference any anomalies in near-real time based on user activity based on trained models

V1: DATA PREP FOR INFERENCING

- Use Case: Detecting anomalies with user’s activity
- Inferencing flow from 10k feet
- Started with python scripts for windowed aggregation

Learnings: Hard to scale for near real time. AD platform runs inferencing every 3 mins as we are impacted by speed of data processing.
V2: IMPROVING DATA PREP PERFORMANCE

- V2: To improve performance, we started using Presto with data on S3 in JSON format
- Live data will be streamed from Kafka to S3. We use Presto for our data warehousing needs
- Presto is an open-source distributed SQL query engine optimized for low-latency, ad-hoc analysis of data*

Learnings: Presto with Parquet has best performance but we need to batch data at 30 secs interval. So it’s not completely real time
GPU ACCELERATION
Accelerate the pipeline, not just deep learning!

- GPUs for deep learning = proven
- Where else and how else can we use GPU acceleration?
  - Dashboards
  - Accelerating data pipeline
  - Stream processing
  - Building better models faster
- First: GPU databases
1+ BILLION TAXI RIDES BENCHMARK

Source: MapD Benchmarks on DGX from internal NVIDIA testing following guidelines of Mark Litwintschik’s blogs: [Redshift, 6-node ds2.8xlarge cluster](https://marklit82.com) & [Spark 2.1, 11 x m3.xlarge cluster w/ HDFS](https://marklit82.com)
MAPD + IMMERSE VS ELASTIC + KIBANA

MapD Core

- Very fast OLAP queries
- JIT LLVM query compiler
- GPUs for compute
- CPUs for parse + ingest
- Limited join support (for now)
- Concurrency?

Immerse

- c3/d3 + crossfilter = nice dashboards
- Backend rendering

Elastic + Kibana

- Fantastic for complex search
- Scales easily (up to a point)
- Indexing consumes more storage (~4-6x)
- Kibana for KPI dashboarding?
ARCHITECTURE
V2 (with MapD)
MAPD VS KIBANA
Dashboards Comparison + Performance Test Method
DASHBOARD PERFORMANCE
MapD Immerse vs Elastic Kibana

Time to Fully Load (seconds)

Days of Data

MapD Immerse (DGX)
MapD Immerse (P2)
Elastic Kibana

< 9s
< 12s
V3: Data Prep using GPU acceleration

- V3: Explored GPU databases like MapD to improve the performance for querying on streaming live data
- MapD offers constant query response times
- MapD has some SQL limitations. We use Presto as an interface & built a “MapD-> Presto” connector for full ANSI Sql features

**GPU Database Performance**

![Graph comparing execution times for different data formats and connectors.](image-url)

- **Execution Time (seconds)**
  - **PRESTO ON JSON**: 20, 25, 30
  - **PRESTO ON PARQUET**: 4, 6, 8
  - **MAPD**: 0.1, 0.1, 0.1
  - **PRESTO + MAPD**: 1.2, 1.2, 1.2

- **Legend**:
  - 10 mins
  - 30 mins
  - 60 mins
FUTURE
EXPAND GPU USAGE
More Data, Less Hardware

Scaling up and out with GPU
EXPAND GPU USAGE
Internal Logs to Cyber Security

Cyber Security Analytics Platform
GOAI ECOSYSTEM
End To End Data Science

A Python open-source just-in-time optimizing compiler that uses LLVM to produce native machine instructions.

Dask is a flexible parallel computing library for analytic computing with dynamic task scheduling and big data collections.
**GOAI ECOSYSTEM**

End To End Data Science

Graph Analytics & Visualization

A CUDA library for graph-processing designed specifically for the GPU. It uses a high-level, bulk-synchronous, data-centric abstraction focused on vertex or edge operations.

H2O4GPU Roadmap

**Graph Analytics & Visualization**

SIEM attack escalation

Dropbox external sharing logs
BETTER DATA PIPELINES
User Defined Functions at Scale

https://github.com/gpuopenanalytics
libgdf, pygdf, dask_gdf
BETTER DATA PIPELINES

HIVE to BlazingDB
BETTER DATA PIPELINES

More Models

https://github.com/h2oai/h2o4gpu

# edges = E * 2^S ~ 34M
JOIN THE REVOLUTION
Everyone Can Help!

APACHE ARROW
https://arrow.apache.org/
@ApacheArrow

APACHE PARQUET
https://parquet.apache.org/
@ApacheParquet

GPU Open Analytics Initiative
http://gpuopenanalytics.com/
@Gpuoai

Integrations, feedback, documentation support, pull requests, new issues, or donations welcomed!
THANK YOU!

NVIDIA Team
Ed Clune
Ayush Jaiswal
Keith Kraus
Rohit Kulkarni
Jarod Maupin
Nixs Nataraja
Rohan Somvanshi
Michael Wendt