S9758 - Machine Learning Based Network Fault Management with Streaming Telemetry Data

GPU Technology Conference 2019

Vladimir Yashin
Senior Data Scientist, CX Platforms
Mar 20, 2019

GPU TECHNOLOGY CONFERENCE
SILICON VALLEY
MARCH 18-21, 2019

About

Vladimir

Yashin

Network engineering

Data Science

- 5 years of network engineering (Service Provider)
- 3 years of building ML solutions
- Cisco CX
- Based in Brussels, Belgium

vyashin@cisco.com

Expanding the Network

Why Telemetry?

SQUID for AD in Telemetry

Beyond RNN

Conclusions

Big is getting Bigger (2021)

Enterprise networks to grow by

75%

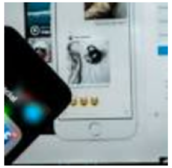
IP traffic to

Nearly double

Incidents to increase **x25**

400/h to 10,000/h

Cost of a failure grows too



CBBC Newsround

Facebook and Instagram suffer most severe **outage** ever
BBC News - il y a 8 heures
Facebook has yet to offer an explanation for the **outage**. ... Buenos Aires-based designer Rebecca Brooker told the **BBC** the interruption was ...

Instagram, YouTube, WhatsApp: Why was some of the internet broken ...
CBBC Newsround - il y a 1 heure

Facebook says technical issues with its family of apps resolved
a 4 heures

records, no indication what started it

urs; worst **outage** in history

HOME > NEWS > CABLING

Network outage grounds passengers at Gatwick Airport

An issue with a Vodafone cable wreaks havoc at one of the UK's busiest airports

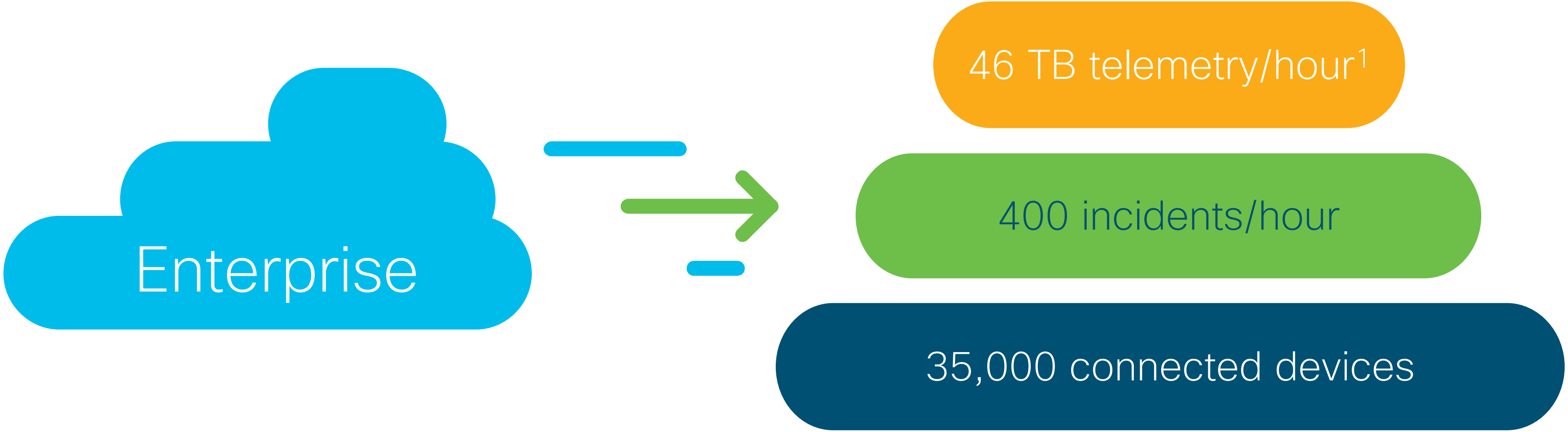
August 22, 2018 By: Max Smolaks

CenturyLink outage takes down several 911 emergency services across the US

Downtime caused by network issue affecting 15 of CenturyLink's data centers.

December 28, 2018 by Catalin Cimpanu

Today

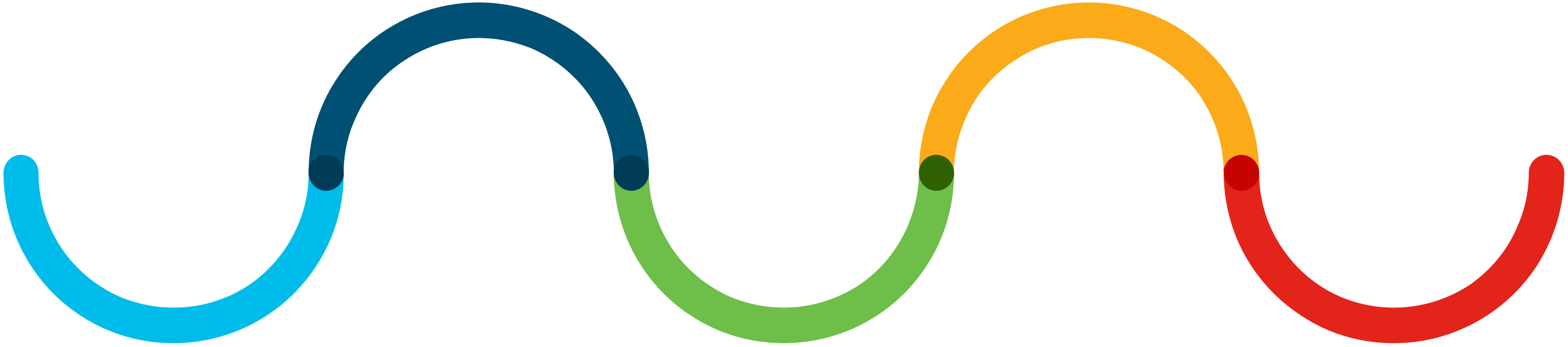


[1] Big enterprise streaming 100+ telemetry streams with 5s interval from 35,000 devices

Autonomous requires Technology Transformation

Massive
amount of data

Distributed processing
Edge + Cloud computing



Node failures
are business as usual

Wire-speed
decision making

AI/ML-powered
Analytics and Automation

Expanding the Network

Why Telemetry?

SQUID for AD in Telemetry

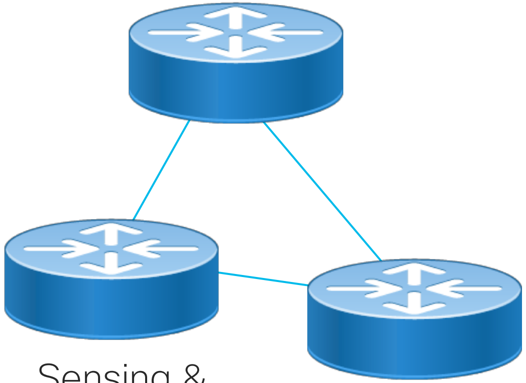
Beyond RNN

Conclusions

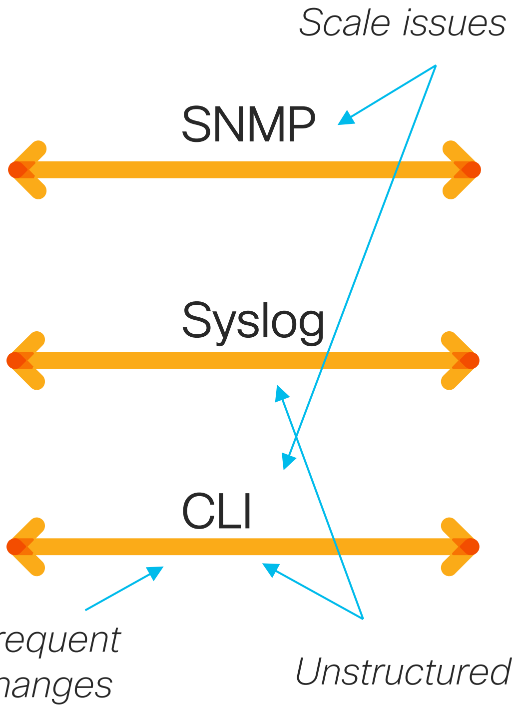
Traditional monitoring is showing its age

No longer suited for Cloud-Scale Network Operations

Where data is created



Sensing & measurement



Where value is

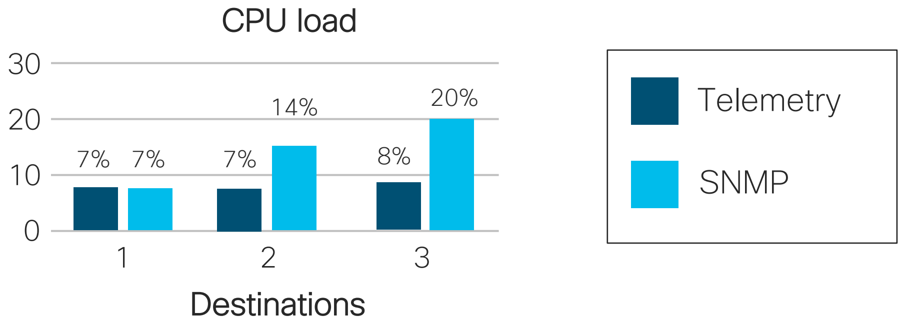
Data processing burden on back-end

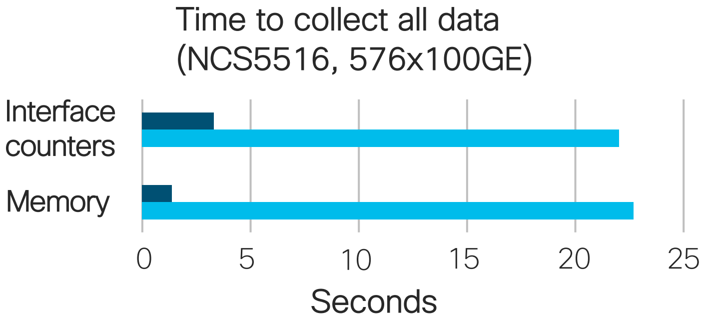
Normalize encodings, transports, data models, timestamps etc.

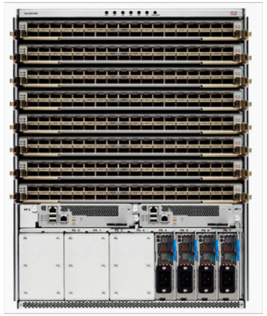
Streaming Telemetry principles

- Data is “pushed” by device towards collector
- Superior scalability
- Standard encoding (Protobuf, JSON) and transport (gRPC)
- Vendor-neutral, model-driven network management designed by users

Pushing more data really does work better







- More streaming data
- Reduced CPU load
- Faster collection

Current challenges

Current challenges	Different types of sensors and telemetry data formats	Complex emergent behavior make simulations and generating synthetic data hard	Asynchronous data sampling rate and arrival	Scalability: Millions of devices and streams in IoT networks	Latency: Need for real-time processing and anomaly detection
Solutions	Great progress in standardization with OpenConfig	Experiments and ML model training on real data from day 1. Requires huge data lakes.	Data buffering and aggregation	Moving model inference and training to edge	Moving model inference and training to edge

- Expanding the Network
- Why Telemetry?
- SQUID: Deep Learning System for Anomaly Detection in Streaming Telemetry
- Beyond RNN
- Conclusions

SQUID ML Pipeline

LSTM Model Gen µService

Telemetry Metric 1
Telemetry Metric 2
....
Telemetry Metric N

Multi-Stage LSTM

Feature Generation
Math Operators
Time Windows

LSTM
LSTM
Fully Connected Layer
Fully Connected Layer

Forecast outputs and compute residues

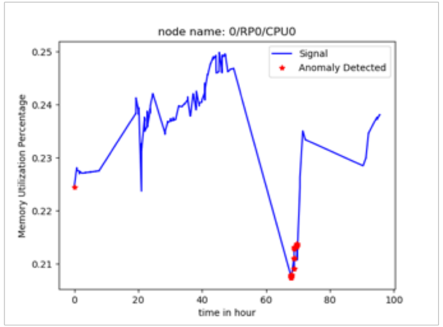
LSTM AD Model

Anomaly Classifier µService

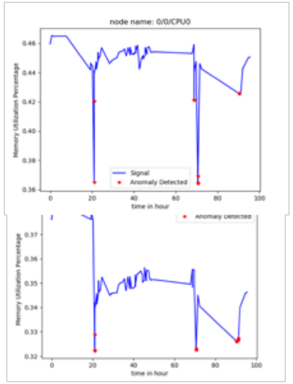
Telemetry Metric 1
Telemetry Metric 2
Telemetry Metric N

Anomaly if residue above threshold
LTSM AS Model

Single-Metric Anomaly



Cross-Metrics Anomaly



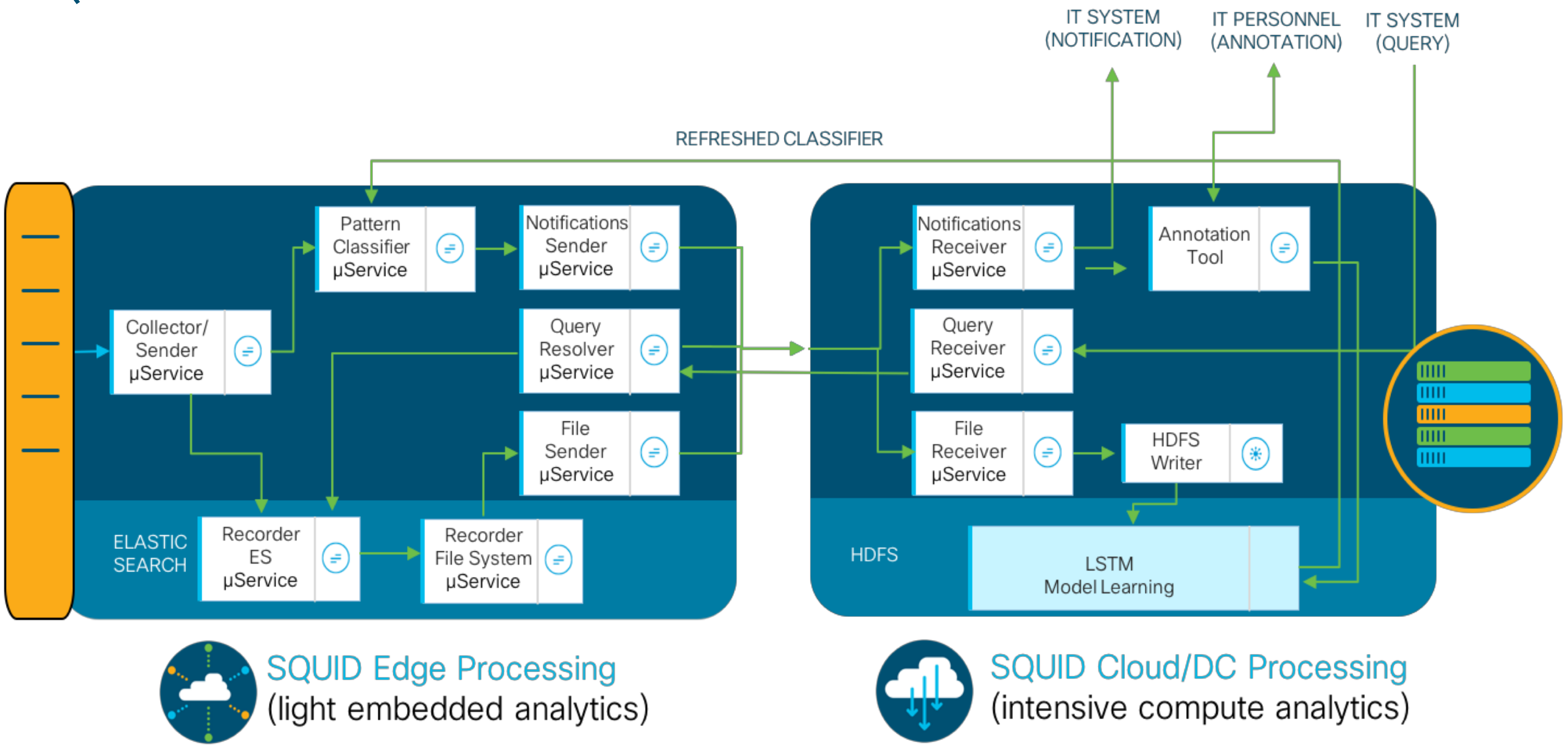
Training (Cisco Cloud)

- 3x Nvidia DGX-1 (8x Tesla V100)
- 15min per model on single GPU
- 3 hours to train all ~300 models to convergence

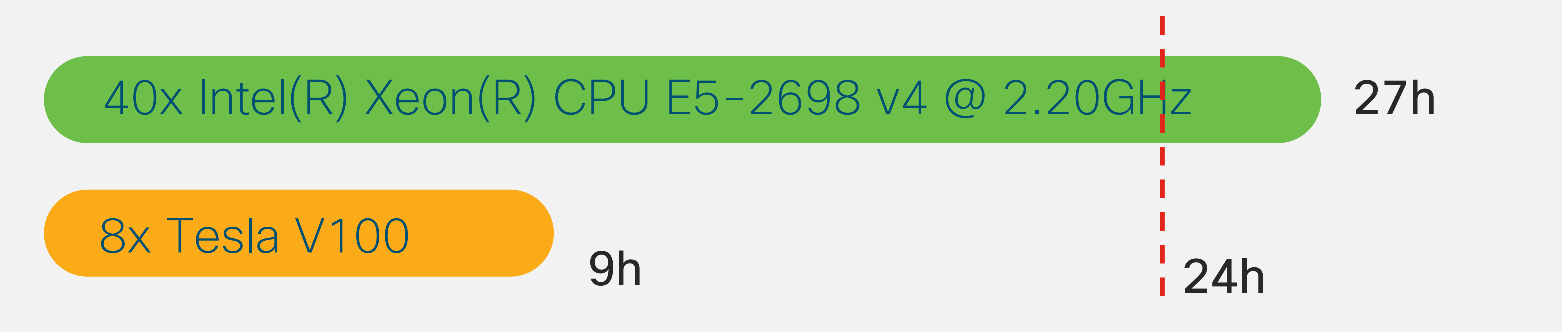
Training/inference (Edge)

- Cisco UCS C480 ML M5
- 8x Nvidia Tesla V100 32GB

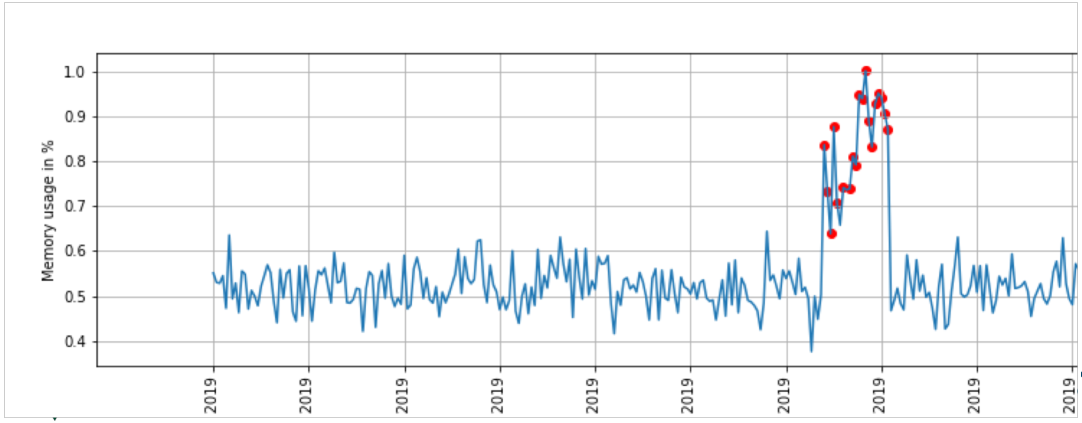
SQUID: Functional Architecture



Training time comparison



Training time of all ~300 models on DGX1 (PyTorch) (CPU vs. GPU)



Results

Data Scientists create models based on SME input	Self-service portal
Model training time: 15min	Real time evaluation
Highly accurate	Reduced accuracy is acceptable
Production-grade	Draft modelling
Highly accurate blackbox (LSTM-based)	Transparent, inspectable model (Linear)

Expanding the Network

Why Telemetry?

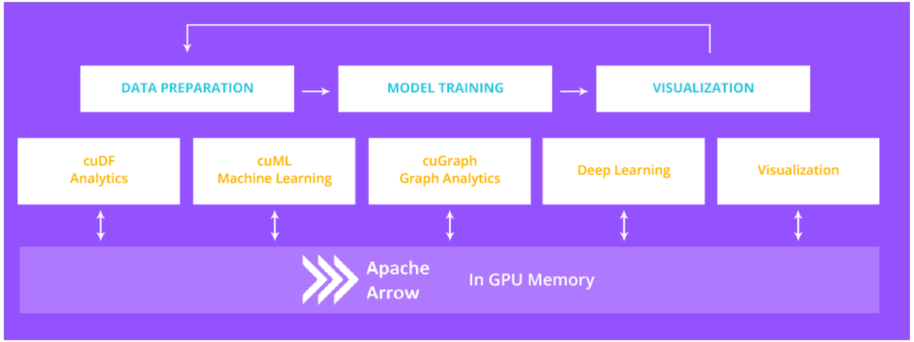
SQUID for AD in Telemetry

Beyond RNN

Conclusions

Nvidia RAPIDS Introduction

- Pandas-like DataFrame in vRAM
- Similar to H2O4GPU
- Vendor-backed
- Can share data with PyTorch and Chainer
- Claims 4x+ speedup over xgboost



- You may find these sessions useful:

S9801 - RAPIDS: Deep Dive Into How the Platform Works
S9577 - RAPIDS: The Platform Inside and Out
S9793 - cuDF: RAPIDS GPU-Accelerated Data Frame Library
S9817 - RAPIDS cuML: A Library for GPU Accelerated Machine Learning

Dataset

Data

- Synthetic Telemetry stream
- One sample per minute
- 43200 samples/device
- 1000 IP routers
- 70/10/20 split

Features

- Time of day
- Day of week

ETL performance

RAPIDS v0.5.1 CUDA 10.0, 1x Tesla V100

```
[3]: train_cudf_csv = cudf.read_csv('training.csv', lineterminator='\r')

[4]: def transform(cpu_mean, cpu_var, data_cudf):
         df = data_cudf
         df = df.one_hot_encoding('hour', 'h', range(24))
         df = df.one_hot_encoding('weekday', 'wd', range(7))
         feature_columns = [c for c in list(df)
                            if c.startswith('h_')
                            or c.startswith('wd_')]
         Y = (df.total_cpu_one_minute - cpu_mean) / np.sqrt(cpu_var)
         return df[feature_columns], Y

[6]: %%timeit -n10 -r3
     cpu_mean, cpu_var = train_cudf_csv.total_cpu_one_minute.mean_var()
     train_X, train_Y = transform(cpu_mean, cpu_var, train_cudf_csv)

437 ms ± 17.2 ms per loop (mean ± std. dev. of 3 runs, 10 loops each)
```

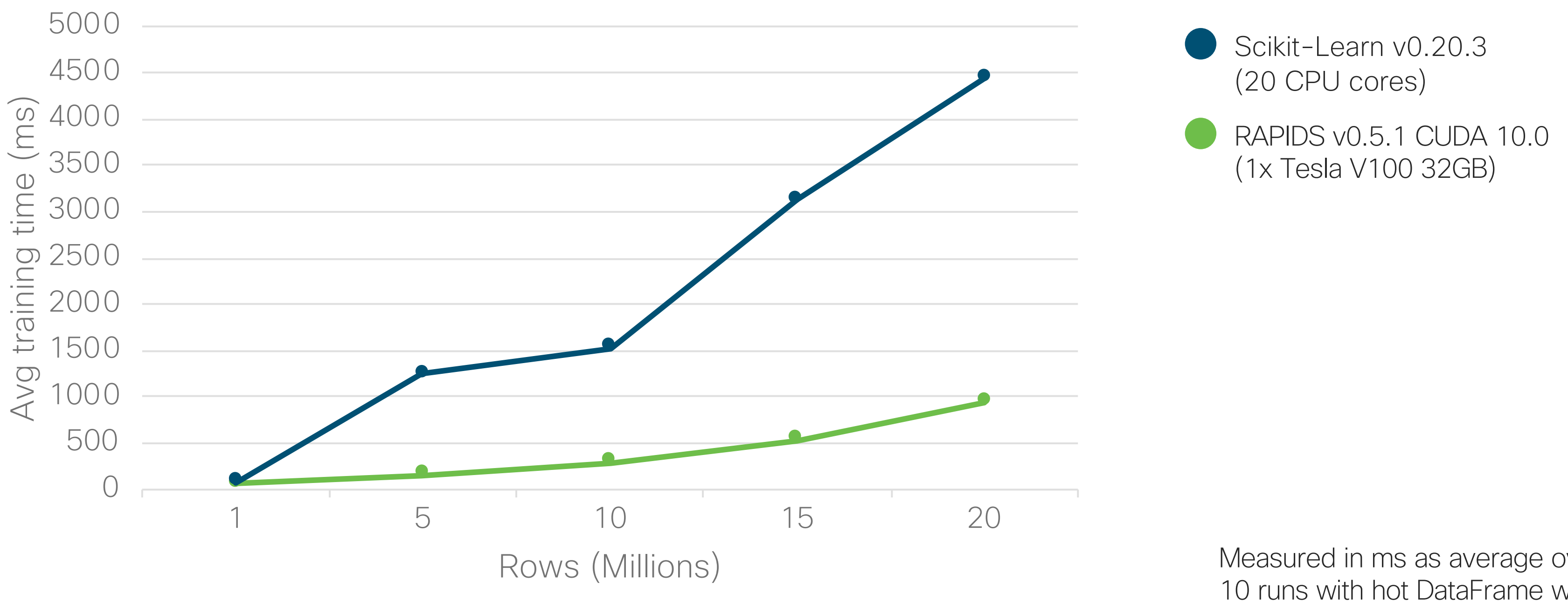
Sklearn v0.20.3, 20 CPU cores

```
[20]: train_df_csv = pd.read_csv('training.csv')

[21]: transformer = ColumnTransformer([
          ('cat', OneHotEncoder(handle_unknown='ignore'), ['hour', 'weekday
          ('num', StandardScaler(), ['total_cpu_one_minute'])
      ])

      def transform(transformer, data_df):
          matrix = transformer.transform(data_df)
          return matrix[:, :-1], matrix[:, -1].toarray()

[22]: %%timeit -n10 -r3
      transformer.fit(train_df_csv)
      train_X, train_Y = transform(transformer, train_df_csv)

18.6 s ± 4.85 ms per loop (mean ± std. dev. of 3 runs, 10 loops each)
```

Dataset 28M rows, 32 column

42x speedup!

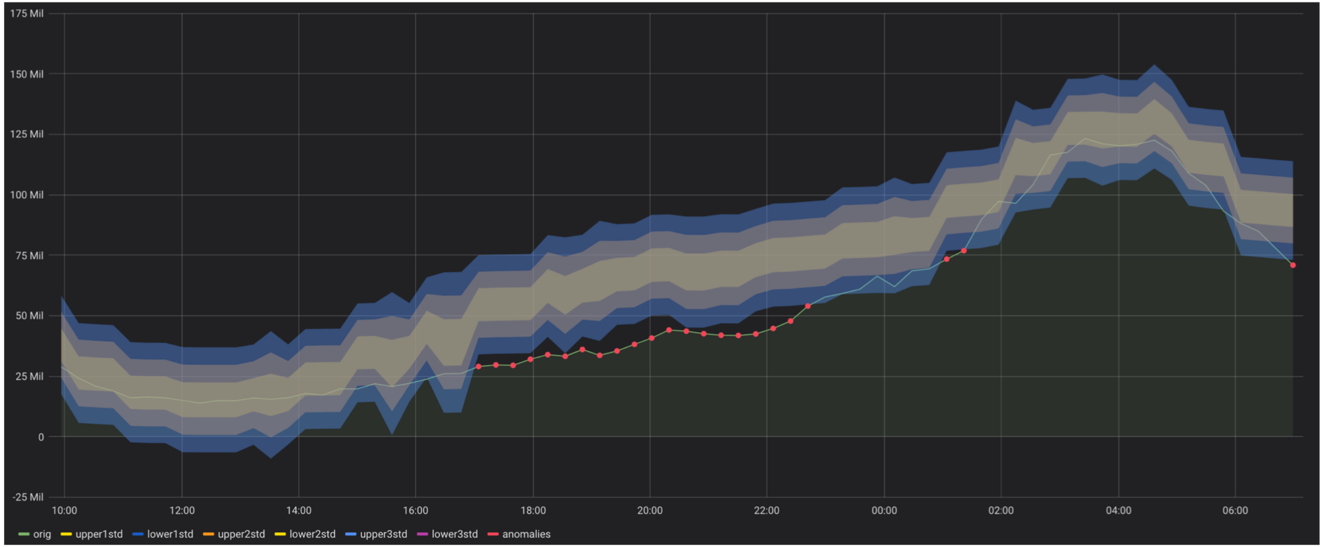
Model training performance



Measured in ms as average over 10 runs with hot DataFrame with 31 columns + 1 response variable.

Linear modelling on Telemetry data

- GLM on RAPIDS is 2x faster than sklearn (~500MB of telemetry from 1 router)
- Still, we are talking *seconds*



RAPIDS: Observations

ETL

- Dataset took 8+ GB of VRAM (double precision?)
- 29GB of VRAM watermark in ETL task (external+internal copying)
- Sklearn is underutilizing CPU in ETL (very short bursts of activity) – this can be tuned/optimized
- Sub-second ETL pipeline means **fast enough for real-time UI**

Training

- Linear regression is way faster on GPU, but this is clearly not a bottleneck in end2end workflow.
- GBM or compute-intensive clustering would've been a whole different story

Expanding the Network

Why Telemetry?

SQUID for AD in Telemetry

Beyond RNN

Conclusions

Conclusions

- Customers prefer integration into existing workflows:
  - Alerts/thresholds instead of « unmanned operations »
  - Inspectable *risk scores* and *weights* instead of magic black box
- Gain trust with simple solutions, deploy DL when customer is ready
- « Give a man a fish… » – huge gains from SMEs' participation

Future work

- RAPIDS memory tuning
- Kubeflow/RAPIDS integration for unified training/production pipelines

Questions?

We value
Your Feedback

Fill in session survey in your GTC mobile app

https://live.eventbase.com/app-download?event=gtcsiliconvalley2019

GET IT ON Google Play

Download on the App Store

Thank you

cisco