Autonomous Driving with Airflow
Where big-data meets high performance computing

Amr Noureldin – Solution Architect
Michal Dura – Big Data Engineer

© 2019 DXC Technology, Proprietary and Confidential
The world’s leading independent, end-to-end IT services company

**Scale & Skills**

- **$21+B** Annual revenue
- **134K** Employees
- **25%** Skilled in next-gen
- **70+** Countries
- **DXC dynamic talent cloud**

**Customer Intimacy**

- **Manufacturing & Automotive**: 16%
- **Healthcare**: 11%
- **Insurance**: 11%
- **Banking**: 14%
- **Retail**: 12%
- **Energy**: 12%
- **Other**: 3%

Excellent client coverage across the globe ...

- **~6,000** Clients
- **200+** F500 clients
- **36** NPS

... enhanced through world-class partner network

**DXC Value for AD**

- **Accelerate time to market**
- **Reduce cost and risk**
- **Improve market leadership**

**Technology-Driven Innovation**

- **$4B** Digital revenue
- **250+** global partners
- **14** strategic co-investing partners

**Streamlined offerings**

- **96** Offerings
- **9** Streamlined Offering Families
- **250+** Industry partners

© 2019 DXC Technology, Proprietary and Confidential

July 16, 2020 2
# Autonomous Driving

<table>
<thead>
<tr>
<th>Level - 0</th>
<th>Level - 1</th>
<th>Level - 2</th>
<th>Level - 3</th>
<th>Level - 4</th>
<th>Level - 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>DRIVER</td>
<td>FEET OFF</td>
<td>HANDS OFF</td>
<td>EYES OFF</td>
<td>MIND OFF</td>
<td>PASSENGER</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>No Assistance</th>
<th>Assisted</th>
<th>Partially Automated</th>
<th>Highly Automated</th>
<th>Fully Automated</th>
<th>Autonomous</th>
</tr>
</thead>
<tbody>
<tr>
<td>Human</td>
<td>Transfer of responsibility</td>
<td>Machine</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
R&D in Automotive Industry – Capabilities are Changing

Extend & reuse current mechanic engineering capability combined with artificial intelligence, software building and IT is important to survive
Requires an end-to-end data and AI capability ecosystem for AD development

Geographically distributed
R&D teams

AD data & models

Collect/Ingest/Store
Manage, Find & Analyze
Perception & Location
Fusion/Motion Control
Simulation
Verification/Recompute
Test/Road Approval

Need for speed
HIGH PERFORMANCE DATA DRIVEN DEVELOPMENT PLATFORM

MULTI-TENANCY

>200 PB superconverged

>100,000 processor cores

Setup in only 3 months

>200 GPUs

>1.150 m²

>2.3 MW

96 x 100 Gpbs to ADC

FACTS & FIGURES
## Technology Stack

<table>
<thead>
<tr>
<th>Business Organizations</th>
<th>Visualizations Collaboration</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Jupyter, Confluence, Jira Software</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Analytics Deep Learning</th>
</tr>
</thead>
<tbody>
<tr>
<td>TensorFlow, KubeFlow, Spark, Apache Drill, hadoop</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Abstraction Orchestration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apache Airflow, Apache HBase, API, Kafka, API, YAML, Azure Pipelines</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Storage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ceph, Data Fabric on Kubernetes, MAPR-DB, MAPR-ES, MAPR-XD</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Virtualization OS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kubernetes, Openshift, Red Hat</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Infrastructure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hewlett Packard Enterprise, Dell EMC, NVIDIA, Azure</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Data Ingest</th>
</tr>
</thead>
<tbody>
<tr>
<td>MapR-ES, Hadoop</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Data Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>DXC</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Security</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grafana, kibana, Jenkins, Nexus, Azure DevOps, SonarQube, Beats, Logstash, Elasticsearch, Prometheus, Active Directory, LDAP, Secure Cluster, Monitoring Deployment Automation</td>
</tr>
</tbody>
</table>

© 2019 DXC Technology, Proprietary and Confidential
Autonomous Driving with Airflow – In a Nutshell (1/3)

- Airflow on OpenShift
  - 1 scheduler instance – big risk
  - Multiple webservers – load balanced (Rest API Calls)
  - Numerous workers – multiple queues

- Automated deployment via Helm charts
  - Various configurations for different instances (also different Airflow versions)
  - History tracking via version control

- On-demand Airflow instances (ex: development purposes, isolated testing – on a service level)
Autonomous Driving with Airflow – In a Nutshell (2/3)

• Integration with the Data and Storage Platform – MapR
  • Loading DAGs from different locations
  • Loading job configurations, used by the different operators

• Integration with the Compute Platform – OpenShift
  • KubernetesPodOperator
  • KubernetesExecutor
Autonomous Driving with Airflow – In a Nutshell (3/3)

- Metrics Collection & Monitoring
  - StatsD → Prometheus → Grafana

- Log collection and aggregation: ElasticSearch + Kibana

- Large scale orchestration: aiming at orchestrating jobs at the scale of 100,000’s / month
  - Ingestion, simulation, reprocessssing, machine learning, …etc
  - Complex DAG dependencies
Apache Airflow - Robotic Drive orchestrator
Platform Orchestration Requirements

- Open Source
- Scalability
- Easy to adapt / extend
- Active community
What do we Orchestrate?

- Data Ingestion
- Machine Learning
- Reprocessing
- Simulation Jobs
Journey from PoC to Production
Airflow at Robotic Drive – the beginnings

- Initial work started at Q2 2019
- Airflow 1.10.2 with CeleryExecutor
- PostgreSQL 9.4
- RabbitMQ
Technical Challenges and Lessons Learned

- Airflow stress and scalability tests

- Bottlenecks in the Architecture:
  - PostgreSQL connection scalability: directly proportional relationship between number of running tasks and database connections
  - Scheduler configuration & performance
Tailor-made Solutions
Operators / Hooks Customizations (1/3)

Customization and standardization of SparkSubmitOperator

**Included „properties_file” in operator constructor**

- Spark application configuration can be provided via separate properties file
- It allows submitting jobs via Airflow, in the same fashion as submitting them standalone from the Hadoop cluster

**Extend list of parameters where templating is supported**

- Better DAGs reusability
- Reduced code duplication
Operators / Hooks Customizations (2/3)

Enrich Airflow logs by adding YARN application details for Spark jobs

- Correlation between Spark application (triggered by Airflow) and submitted YARN job is challenging to discover when using YARN cluster mode

- Out of the box: YARN Application ID logged in the task logs only when a YARN job fails

- Extension: Extract and log the following for all Spark tasks:

  1) YARN Application ID
  2) YARN Tracking URL
  3) Diagnostics (Failure root cause)
Operators / Hooks Customizations (3/3)

YARN application details visible in Airflow logs for Spark jobs

1. Submit Spark application with custom YARN tag
   - SparkSubmitHook
   - Hadoop Cluster

2. Query YARN RM REST API and get application details
   - SparkSubmitHook
   - Hadoop Cluster
Custom Authentication Methods

LDAP Secured REST API

- REST API usage allowed only for dedicated AD role
- Complete integration with LDAP
- Only one role specified for REST API – no separation between endpoints so far
Airflow in Production
Production-ready Airflow instances

Robotic Drive supports by default 3 main instances used in the platform:

- **Development** (used mostly to test new Airflow deployments / features)
- **Staging** (testing new DAGs)
- **Production** (for full production usage)

Current stable setup is created using **Airflow 1.10.10** and **Celery Executor**
User-based Airflow instances

- Deployment automated via Helm Charts
- Airflow created as a Kubernetes project

These 2 points makes it possible to fully parametrize Airflow deployment and create many Airflow instances on-demand. In the Robotic Drive Platform each user is able to create their own, separate Airflow instance.

It helps to eliminate problems related to testing new DAGs, Operators and other features and changes that might impact other users, especially when multiple developers are working on the same component.
Monitoring Airflow
Monitoring Airflow
Monitoring Airflow
Monitoring Airflow

Dag Processing Last Duration

0 250 500 750 1.0 K 1.3 K
7/11 12:00 7/11 16:00 7/11 20:00 7/12 00:00 7/12 04:00 7/12 08:00 7/12 12:00 7/12 16:00 7/12 20:00 7/13 00:00
What’s next?
Looking forward to…

- Airflow 2.0: HA Scheduler + Performance Optimizations
- Advanced Authentication + Authorization
- Extend and stabilize monitoring metrics
- Stable API vs Experimental API