CLOUDERA

WHITE PAPER

WINNING IN AUTONOMOUS DRIVING WITH CLOUDERA DATA PLATFORM

Understanding the Complexity of the Autonomous Driving Data Loop Promotes Success

CLOUDERA
DATA
PLATFORM

Table of Contents

150

The average amount of data needed to build an autonomous vehicle is estimated to be 150PB.

Introduction

The march towards autonomous vehicles continues to accelerate, now breaking into a slow run. While expert opinion differs on the specific timing and use cases that will emerge first, few deny that self-driving cars are in our future. In fact, more than one hundred cities and dozens of vendors worldwide are testing autonomous vehicles today in a race that will disrupt the automotive and technology landscape.

Car companies, parts manufacturers as well as new contenders from the IT sector alike pursue the challenge of "teaching" vehicles to drive themselves in a failsafe manner. The staggering amounts of data and machine learning tasks inherent to building autonomous vehicles require car-builders to add computer sciences and deep learning to the very core of their businesses.

This white paper presents key technologies that Cloudera contributes to streamline all steps of the autonomous driving development lifecycle into a scalable AI-enabled Enterprise Data Cloud.

Cars Will Learn

While it would be easy to consider autonomous vehicle development a classical automotive engineering challenge, in reality it is much more of a data analytics and machine learning endeavor. In fact, deep learning is at the core of the perception layer of autonomous vehicles. Whether detecting objects or segmenting images from autonomous car camera feeds, predicting an object's path or classifying the current traffic situation as dangerous, the car leverages machine learning models that have previously been trained during similar situations to abstract and adapt to new situations and make decisions. Model training is foundational technology where the quality and diversity of data is key to the model's success.

A Petascale Endeavor

Machine learning models, particularly Neural Networks, perform better and more accurately when they are trained on larger datasets of great variety. Such performance is critical as both autonomous driving engineers and industry regulators demand extreme reliability and precision. As a result, huge collections of autonomous vehicle data, describing virtually any condition a vehicle may encounter, must be captured to train and continuously improve autonomous driving models. To facilitate this reality, autonomous test vehicles continuously capture huge volumes of test vehicle data—this quickly results in a petascale endeavor.

Most participants in the autonomous vehicle research field have inevitably identified the limits of classical data management technologies (i.e. plain file system storage and legacy database technologies) and are looking towards big data technologies to build searchable, petascale data and metadata repositories.

3-5

Experts estimate 3-5 GPU's will be needed in order to run a car's perception layer.

The Autonomous Driving Data Loop

To better understand how data is utilized to teach cars to drive, consider the figure and sections below describing the phases of the autonomous driving data loop.

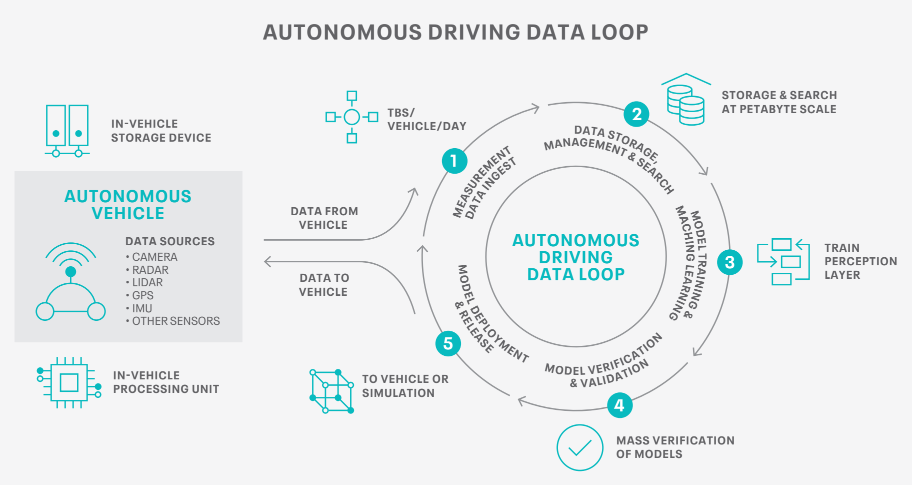

AUTONOMOUS DRIVING DATA LOOP
IN-VEHICLE STORAGE DEVICE
AUTONOMOUS VEHICLE
DATA SOURCES
• CAMERA
• RADAR
• LIDAR
• GPS
• IMU
• OTHER SENSORS
IN-VEHICLE PROCESSING UNIT
TBS/ VEHICLE/DAY
DATA FROM VEHICLE
DATA TO VEHICLE
TO VEHICLE OR SIMULATION
1
MEASUREMENT DATA INGEST
2
DATA STORAGE, MANAGEMENT & SEARCH
STORAGE & SEARCH AT PETABYTE SCALE
3
MODEL TRAINING & MACHING LEARNING
TRAIN PERCEPTION LAYER
4
MODEL VERIFICATION & VALIDATION
MASS VERIFICATION OF MODELS
5
MODEL DEPLOYMENT & RELEASE
AUTONOMOUS DRIVING DATA LOOP

Figure 1. Phases of the Autonomous Driving Data Loop

Phase 1: Measurement Data Ingest

PHASE DESCRIPTION

The autonomous drive teaching process begins on the left side of the diagram above, with autonomous test vehicles, copiloted by humans. The primary function of these vehicles is to collect huge volumes (dozens of terabytes per vehicle, per day) of real-world driving data from highly accurate reference sensors (camera video, radar, lidar, ultrasonic, GPS, IMU/CANBUS and other external and internal sensors). It is this data, describing virtually every operating condition a vehicle may encounter, that will provide the Ground-Truth inputs to "teach" (or train) a vehicle to drive.

The data collected during this process has traditionally been managed in two steps. First, the data is collected and stored on hard disks residing within the test vehicle. Next, the hard drives are connected to an internal network, where it is then ingested into a centralized environment for subsequent processing.

CHALLENGES

Clearly, the data ingestion process outlined above is inefficient from a data management perspective, requiring redundant data management overhead (i.e. data storage, processing and hardware costs) on both the vehicle and within the enterprise data analysis environment.

SOLUTION

With the impending rise of 5G networks, carmakers and tier-1 suppliers are looking at new methods, such as on-road ingestion to collect, pre-process and transmit test data to storage systems in a streaming fashion (in fact, some companies in that space have already introduced basic capabilities in this area). To facilitate this, two capabilities are required:

- **Next-generation service-oriented gateways**—these devices, traditionally residing within the vehicle itself, have been used to isolate connected vehicle communication channels (i.e. telecommunications, Bluetooth, Wi-Fi, etc.) from the underlying vehicle operational control systems, providing a security firewall to prevent malicious intrusions from adversely impacting vehicle behaviors. Recently however, devices such as NXP's Vehicle Network Processors for Service Oriented Gateways have evolved into computing devices with sufficient compute power to support edge data ingestion and pre-processing workloads.

Insight and challenges

"You see all kinds of crazy things on the road, and it turns out they're not all that infrequent, but you have to be able to handle all of them," said Bryan Salesky, Chief Executive, Argo AI.

"80 percent of the technology needed to put self-driving cars into routine use—the radar, cameras and other sensors that can identify objects far down roads and highways. But the remaining 20 percent, including developing software that can reliably anticipate what other drivers, pedestrians and cyclists are going to do, will be much more difficult."

It's much more difficult to prepare self-driving cars for unusual circumstances —person carrying a surfboard in winter.

- **High-performance data stream processing**—to enable a wide range of anticipated connected vehicle use cases (i.e. predictive maintenance, service recommendations, real-time service and retail offers, etc.), data ingestion and stream processing capabilities are required both at the vehicle "edge" and within Cloud and Data Center environments. Apache NiFi and MiNiFi, components of Cloudera DataFlow (CDF), provide critical capabilities in this area, including the ability to collect and process high volume, real-time streaming data at the vehicle edge and guarantee delivery of data to the Cloud or on-premise data centers. More specifically, CDF provides the following capabilities:
  - Quickly and easily build edge data ingestion flows via an intuitive visual user interface
  - Aggregate, compress and encrypt connected vehicle data
  - Prioritize transmission of vehicle data to the Cloud or Data Center. Flexibly define prioritization rules based on data content and further refine prioritization based on current network volume and bandwidth.
  - Data buffering for automatic handling of system interruptions, network issues, etc.
  - Track data provenance and lineage of streaming data, providing confidence in the origin and usage of data.

Phase 2: Data Storage, Management and Search

PHASE DESCRIPTION

After data ingestion, it must be made available to a wide range of data consumers within the organization. As such, it must be stored and managed as necessary. Critically, given the truly massive amounts of data encountered, relevant data sets must be made searchable for key users (i.e. Machine Learning Engineers, System Test Engineers) to develop the perception layer and train the core deep learning models, as outlined in the next sections.

CHALLENGES

Data Storage—When contending with petabyte levels of data volume, cost effective storage is essential. Two considerations become critical—managing the trade-off between data access performance and cost through data-tiering (i.e. "hot" versus "cold" data), in addition to optimizing the blend of deployment options (i.e. on-premises, multi cloud and hybrid cloud deployment methods).

Search—Given the volume of autonomous drive data, search technology becomes both a mission-critical capability and competitive advantage for participants in Advanced Driver Assistance System (ADAS) and Autonomous Drive (AD) system development. To grasp this, consider the daunting task of a Machine Learning Engineer or System Test Engineer trying to find relevant data among the petabytes of autonomous vehicle test data. Specific search-related use cases span the learning lifecycle including:

- **Model training**—Data scientists require the ability to search for and identify very specific training and test data sets from a huge population of input data sets collected from around the world. In a big picture, data scientists might search for all data created in snowy conditions, or conditions with high bicycle traffic interspersed with automotive traffic. This search precision prevents providing either too little or too much (unrelated) data to model training exercises. Specifically, search assists in optimizing models to detect and improve upon corner-case scenarios, including situations that current models are not able to handle today (e.g. the difference between someone riding a bicycle and someone holding a bicycle—which have totally different behavior in traffic).

Interesting facts

Most of the legislation enabling true autonomous vehicles on public roads is not written. Practitioners expect that they have to quickly drill down on scenarios via search technology (ie: a truck approaching a pedestrian crossing with more than 3 pedestrians in it). The engineers will need to prove that their perception layer works on 9,999 of 10,000 instances of those types of scenarios.

- **Model verification and validation**—During the verification pipeline (outlined later), search can swiftly select only the relevant scenes from driving data, drastically reducing time-to-market and improving overall test-coverage while allowing more bandwidth for long-running regression testing, etc.
- **Visual inspection and summary statistics**—Often it is simply necessary for analysts and engineers to eyeball the data. But doing this without search is like browsing the internet without Google.

SOLUTION

Cloudera Storage Solutions—Cloudera embraces the modularization of storage and compute resources extending to petascale datasets. By utilizing enterprise-grade, cost-effective, distributed storage (incorporating HDFS and Kudu), while integrating with common distributed storage systems on the cloud (i.e. Azure ADLS, Amazon S3), Cloudera provides a comprehensive storage solution supporting both on-premise and cloud deployments.

In addition to our current capabilities, Cloudera will soon go commercial with Apache Ozone (beta preview), a scalable, redundant, and distributed object store for Hadoop. Apart from scaling to billions of objects of varying sizes, Ozone supports very dense server configurations. The Ozone roadmap includes support for the Network File System (NFS) protocol and POSIX compliance as well as a storage backend for LUN-like container storage.

Equally important is the interoperability of distributed compute resources with the various storage platforms in question. Since Cloudera query systems (i.e. Spark or Impala) are optimized to perform on object stores such as Azure ADLS and Amazon S3, Cloudera provides large-scale data processing on any infrastructure in any location.

Search—Cloudera Search directly provides the technology to index petascale measurement data and make it accessible throughout the autonomous driving lifecycle. For end-users and analysts this means google-like discovery and analysis, while the search API provides the ability to feed applications and machine learning models with the specific episodes required, rather than flooding them with superfluous and repetitive data.

Phase 3: Model Training and Machine Learning

PHASE DESCRIPTION

Training the Perception Layer—How well a machine learning model can predict an outcome is determined by the amount, diversity and quality of the training data and the machine learning algorithms employed. To train the autonomous drive perception layer, vehicle measurements must initially be labeled by a human to provide trusted ground truth. For example, to facilitate machine learning for image recognition, humans manually map pixels within a frame to a set of object classes (i.e. a car, person, street, sidewalk, etc.). This process is, of course, extremely tedious and costly. However, while still initially required, developers and engineers are increasingly relying on automated methods for labeling and annotation to perform ground truth estimation, with cross-checking procedures to ensure consistency with the high-precision reference sensor data collected from within their fleets. In addition, modern practices utilize machine learning itself to determine specific data points within the overall data population that require more and better labels to improve model accuracy, as documented in a recent publication by Cloudera Fast Forward Labs.

Once the quality of the labels is deemed to be the "ground truth", ML engineers then use them to train machine learning models (such as convolutional neural networks) to detect and localize objects within images with pixel-level accuracy. Such an image classifier is, of course, just one element of the car's perception layer which combines information from multiple sensors and their corresponding machine learning algorithms at runtime.

To reach acceptable accuracy and performance, ML engineers need to continuously train, test and iterate on model training. When the ML team obtains an initial level of confidence, the model will be subject to significantly more testing during the mass verification phase (covered below). This process is repeated for each subsequent change and new version of the model.

CHALLENGES

Managing the Machine Learning System Environment (Traditional)—Due to the fact that machine learning operations generally take place within one or many specialized machine learning and training environments, several system management challenges arise:

- ML engineers require specialized and isolated machine learning development environments to experiment with a wide range of different ML libraries and versions. This often requires supporting dedicated custom docker containers, in addition to classic laptop data science environments.
- As autonomous vehicle research is a world-wide effort spanning multiple time zones and regions for most companies, the ML engineer needs to share both the code and the environment with others in the global team for collaboration, cross-validation and readying it for production post development. Given the nature of the individualized, customized ML environment, this can be difficult to achieve.
- Finally, as models are deployed into production, they are deployed either as web services or run within dedicated mass-inference environments such as Apache Spark, Apache NiFi or directly on the Edge.
- Anywhere along this road, ML engineers may require more resources (GPU, RAM, CPU) than available on their customized local environments, which simply may not be possible due to system limitations.

Consequently, in the absence of advanced system environment management capabilities, it can be extremely difficult to optimize machine learning and training environments, even when grouped into a cluster of GPU-enriched servers. For example, GPU and CPU resources dedicated to individual ML engineers often end up idle, resulting in poor system utilization. More generally, ensuring that resources are pooled and shared fairly among all parties can be challenging indeed. Other issues commonly grappled with include:

- How do you provide convenient and high-performance access to training and test data in your data lake from the machine learning environment?
- How do you provide the various software libraries required by the machine learning algorithms on the data lake? Different ML engineers may demand conflicting versions of libraries.

- How can the operations team ensure the same governance standards that apply to all other IT systems (such as the central data lake and smaller databases) apply to the machine learning environment as well? Such considerations include integration points with identity management systems such as LDAP, Microsoft Active Directory, Access Control Lists and permissions management on third-party storage systems or big-data specific systems such as Apache Sentry or Apache Ranger. How are access credentials for those data sources provided in the machine learning environment?

SOLUTION

Cloudera Machine Learning—Cloudera Machine Learning (CML) addresses all of the challenges outlined above, providing a collaborative, yet fully customizable experience for machine learning engineers while enabling easy and secure access to all datasets within the organization and processing resources of a cluster. Machine learning engineers can focus on their job while IT Operations teams can provide clustered GPU and CPU resources in a secure and governed manner. Key CML features include:

- Machine learning environments are represented as projects that can be collaborated upon and shared among groups of engineers They exist as a scalable group of docker containers and are managed in CML's Kubernetes-based runtime. Machine learning engineers build models via a customizable code-based and fully interactive interface and can run different stacks of libraries side-by-side, without version conflicts. Alternatively, they can also use their own preferred editor, which increases acceptance among the engineers.
- CML provides the ability to deploy any model directly as a scalable web service.
- Machine learning engineers can programmatically drill down (search) on the data that is required to optimally train a model via fine-grained queries via Cloudera Search, and provide results directly to training activities regardless of the underlying storage system.
- IT Operations teams can model the organizational structure via full LDAP and Active Directory integration and individual and group quotas on CPUs and GPUs.

With the introduction of the Cloudera Data Platform, operating this interactive training experience becomes even more scalable and efficient: CDP incorporates Cloudera Machine Learning (CML), which provides the same user interface and functionality as the Cloudera Data Science workbench as an elastic, massively auto-scalable service on various Kubernetes distributions in the cloud and on-premise, hence simplifying dependency management of custom-build libraries. Satisfying on-demand requests for machine-learning tasks and model training has never been as simple as this.

Phase 4: Model Verification and Validation

PHASE DESCRIPTION

Mass Verification of Models—Once the perception layer model (created in the previous section) is developed, it must now be deployed into a mass verification environment, where the real-world autonomous vehicle data previously collected and labeled is replayed to test the operation of the perception layer. While engineers ideally expect 100,000 miles of "real world" vehicle testing to certify a new perception layer release as safe, time-to-market requirements dictate the need for a simulation-based approach to testing on very large reprocessing (or verification) clusters. More specifically, the general workflow for mass verification of models includes the following steps:

- **Episode selection**—Test engineers need relevant data to prove a models' safety, but often a test drive produces dozens of Terabytes of homogeneous and thus irrelevant data, for example a drive on an empty highway might safely be excluded if the test focuses on a busy scenario in urban mobility that the drive will lead to.
- **Preprocessing (optional)**—This may include translation into intermediary formats and identifying the right measurement files to form a consecutive drive as well as navigating to the correct sensor groups within measurement files.
- **Reprocessing**—This entails the scheduling and batch orchestration of the actual verification process. Since the form factor of the perception layer is typically a container, this step is executed on a container runtime environment. Since the containers typically run neural networks inference for object detection, this step is also referred to as mass inference.
- **Postprocessing and Performance Measurement**—Involves the distributed evaluation of results. The results are typically called "Object Lists" as they contain the objects that the perception layer detected in the reprocessing step. If, for example, a new version of the perception software would not find a previously detected object it would mean a significant degradation to the new release's accuracy.

Deployment of Machine Learning Models to Testing Environment—The deployment of perception models from the machine learning environments typically require encapsulation within containers such as Docker. However, unlike the machine learning engineer, who develops and tests software interactively and often requires compute capacity spontaneously, the Test Engineer runs massive batch applications against predefined service level agreements (SLAs) that equate to perception software release schedules.

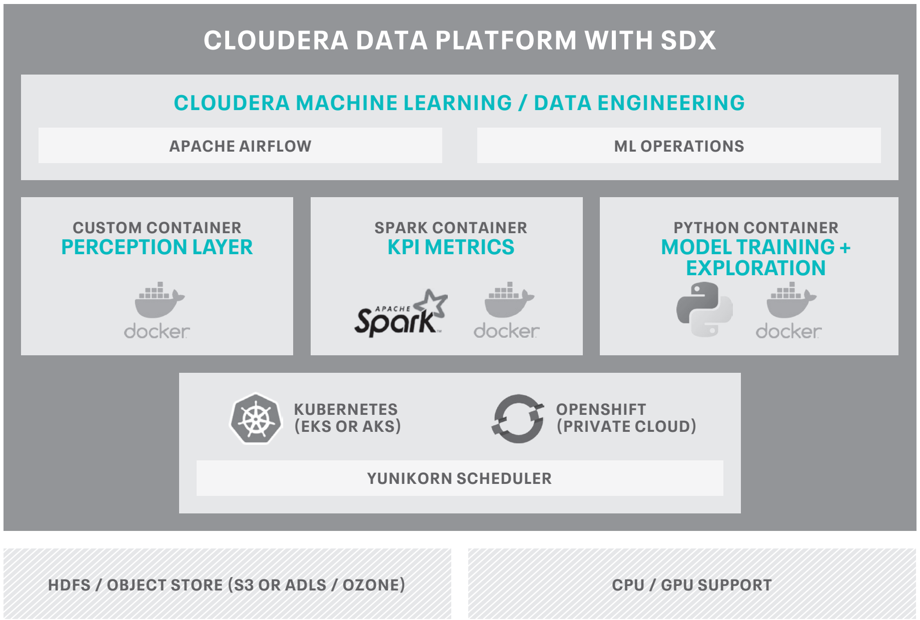
CHALLENGES

Engineers who conduct mass verification of the car perception software require a reliable batch processing environment for Docker containers across hundreds of nodes. This in turn requires scheduling and orchestration. The obvious choice (apart from niche players such as Mesos) for scheduling containers is Kubernetes. While Apache Airflow is gaining popularity for high-level orchestration, there is clearly a gap for ML-centric orchestration and model operation capabilities such as model drift management and feature and model registries. What practitioners are missing is a cohesive enterprise software distribution that combines both of these functions into a meaningful platform product.

SOLUTION

Cloudera provides a comprehensive solution for massively scalable reprocessing environments addressing the challenges described above. Consider the following points:

- Just as a machine learning engineer would leverage Cloudera Search to locate and retrieve the right data for training, the very same search interface can be used to assist a test engineer in the episode selection process, ensuring that models via specific data (i.e. corner-case testing in a specific country) while minimizing data unrelated to the test, improving test relevance and engineering productivity, while reducing time to market.
- To build the reprocessing environment to run a perception layer, Cloudera supports mass inference in Docker containers on Kubernetes via the Cloudera Data Engineering product line, which includes enhanced Kubernetes batch scheduling capabilities as well as advanced orchestration features. Models developed in the prior phase can be orchestrated to run at scale across thousands of instances of docker containers.
- To implement pre and post-processing and performance measurement, Cloudera provides a truly distinguished enterprise experience of Spark on Kubernetes. This includes countless contributions to Spark itself, such as a new shuffle service to increase runtime performance.



With this, Cloudera Machine Learning and Data Engineering support model development, running models at scale as well as evaluating the results via a unified user experience.

Phase 5: Model Deployment and Release

PHASE DESCRIPTION

Tested algorithms are then deployed into either Hardware-in-the-Loop/Hardware-Open-Loop (HiL/HoL) simulation environments or directly into an actual vehicle.

In the case of HiL/HoL, unmounted control units from vehicles are updated with the new software and "replayed" with the same data used in the reprocessing step. In the case of model deployment directly into the vehicle, the in-vehicle processing unit is updated and the algorithm is then tested in real-world driving conditions.

CHALLENGES

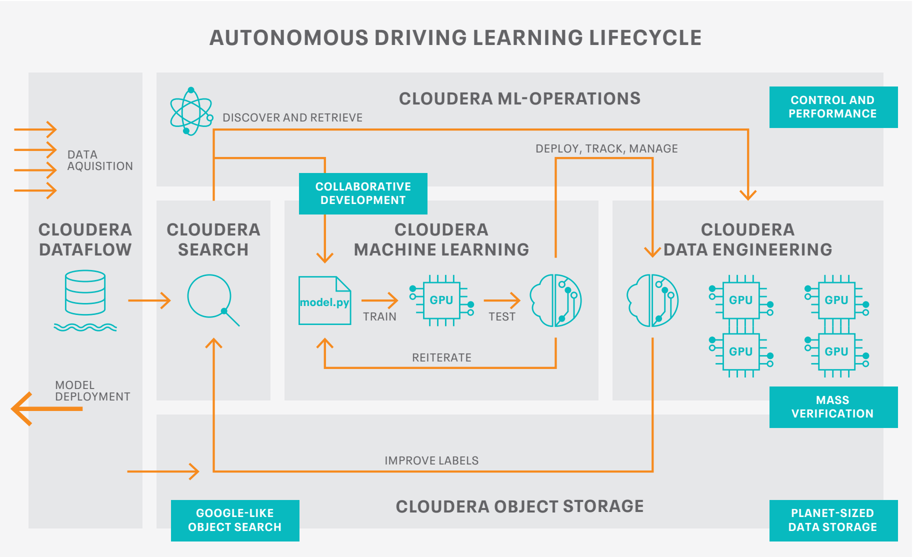
Much like the data ingest process (from the vehicle to the data management environment), the model deployment process (from the data management platform to the vehicle) has traditionally involved a physical "hard-wired" data connection. Obviously, large scale deployment of model updates represent an extremely time consuming and costly endeavor.

SOLUTION

With the impending advent of 5G networks, automakers will increasingly rely on "over the air" update capabilities for the model deployment process. As illustrated below, Cloudera DataFlow (CDF), described previously in section one (Measurement Data Ingest) provides the ability to publish machine learning models, trained and tested in Cloudera Data Science Workbench, directly to Apache MiNiFi agents, where updated models (i.e. TensorFlow) can be used for inference.

Architecture for Success

As illustrated in the diagram below, Cloudera provides a complete open-source solution to enable the Autonomous Driving Learning Lifecycle.



Cloudera's Key Differentiators:

- All data leveraged across the autonomous driving learning lifecycle is available on a unified Cloudera Object Storage layer, which incorporates Apache Ozone on-premise as well as AWS S3 Azure ADLS/ABFS in the cloud.
- Cloudera Machine Learning (CML) Runtime offers an individual and scalable machine learning development environment based on Kubernetes. It enables full Continuous Integration/Continuous Deployment (CI/CD) functionality through sharing of projects, notebooks, containers and source code management integration.
- Models developed in CML can be deployed to Cloudera Data Engineering Runtime for mass verification on Kubernetes in the same containers they were developed in. Alternatively, the CML and the data engineering environment runtimes can also be based on Apache Spark. High-level batch orchestration is achieved via Apache Airflow.
- Cloudera ML-Ops provide the ability to deploy and control models in both a batch and "as-a service" context.
- Cloudera Search can be leveraged for large scale label and data management operations associated with training the perception layer.

Summary and Conclusion

The trend toward autonomous vehicles is clearly underway, and data is a major component and bedrock to its success. Critical to enabling this reality is a high performance, scalable and reliable data management platform that can process the extreme volume, velocity and variety of data that is required, and for it to work at the scale needed to be economically viable. It is only in recent years that this type of technology has become commercially available. Leading automakers, leveraging these capabilities, are driving the rapid ascension of autonomous mobility we are seeing today.

As autonomous vehicle enablement is both highly capital intensive and labor intensive, data management solutions must be chosen for not only today, but for the decades ahead. Cloudera is a leader in delivering this type of next generation data management platform that allows stakeholders in this emerging market to better manage and wring the most value out of data, collaboration and insights derived. As a result, these systems will help deliver self-driving vehicles and technology into the mainstream by ensuring the performance, scalability and reliability required. More importantly, they will help ensure the safety of the passengers in these vehicles and the vehicles around them.

Why Cloudera

- Hybrid and Multi-Cloud—Run your analytics on the clouds you choose. Easily and securely move data and metadata between on-premises file systems and cloud object stores.
- Analytics from Edge to AI—Apply real- time stream processing, data warehousing, data science and iterative machine learning across shared data, securely, at scale on data anywhere
- Security and Governance—Use a common security model, role and attribute based access policies and sophisticated schema, lineage and provenance controls on any cloud.
- 100% Open—Open source, open compute, open storage, open architecture and open clouds. Open for developers, partners, and open for business. No lock-in's ever.

About Cloudera

At Cloudera, we believe that data can make what is impossible today, possible tomorrow. We empower people to transform complex data into clear and actionable insights. Cloudera delivers an enterprise data cloud for any data, anywhere, from the Edge to AI. Powered by the relentless innovation of the open source community, Cloudera advances digital transformation for the world's largest enterprises.

Learn more at cloudera.com

Connect with Cloudera

About Cloudera:
cloudera.com/more/about.html

Read our VISION blog:
vision.cloudera.com

Follow us on Twitter:
twitter.com/cloudera

Visit us on Facebook:
facebook.com/cloudera

See us on YouTube:
youtube.com/user/clouderahadoop

Join the Cloudera Community:
community.cloudera.com

Read about our customers' successes:
cloudera.com/more/customers.html

CLOUDERA

Cloudera, Inc. 5470 Great America Pkwy, Santa Clara, CA 95054 USA cloudera.com