Designing and Building an AI Information Architecture for Faster AI Results and More Data Access Using IBM Storage for Data and AI Solutions with NVIDIA DGX Systems

Contents

Introduction

Artificial intelligence (AI), machine learning (ML), and deep learning (DL) are poised to transform how people and enterprises work and learn. These tools can collect data from edge to core to cloud, analyze the data in near real time, extrapolate from previous results, and infer actions to affect outcomes. This can be applied practically anywhere, such as industrial processes, experiential engagement, healthcare, sales, and real-time decision making.

Current challenges facing data-driven organizations are the need to ingest and analyze an exponentially increasing deluge of data in ever shorter amounts of time as well as supporting more complex models and simulations where, often, actionable insight and responses need to be made in real time. To meet the demands of these current and emerging data challenges, it is necessary to scale out the data center in an efficient way.

Effective AI adoption requires bringing together teams and data in an organized and effective fashion. To simplify adoption of AI, IBM and NVIDIA present this end-to-end reference architecture built specifically to address the needs of wildfire management. The solution is an integrated, individually scalable infrastructure to implement an AI Data Pipeline from ingest to insight. It illustrates a complete AI Data Pipeline solution for the real-life DL use case of training a model to detect smoke for use in prescribed firefighting. The dataset used to train the model was developed from video captured by unmanned aerial systems (UAS) during a controlled burn in Florida.

Although the paper uses the smoke detection use case the model is not specific to smoke detection and can be used by other industries. NVIDIA has led the AI computing revolution, leveraging the power of the modern GPU with its massive processor core count and parallel architecture which is uniquely suited to the massively parallelized operations of ML and DL. Modern GPUs exceed and add to the capabilities of traditional CPU-based architectures. IBM delivers cost effective software-defined storage and hardware solutions with proven scalability to enable massively parallel processing and a path to cost effectively scale from terabytes to petabytes and exabytes of data. IBM offers software-defined storage to help customers navigate this path in a way that allows data scientists and data engineers to focus on AI problems without becoming bogged down in data challenges and cost limitations.

Intended Audience

This paper is intended for enterprise leaders, solution architects, and other readers interested in learning how the IBM Storage for Data and AI with NVIDIA DGX™ systems solution simplifies and accelerates AI.

The data acceleration (caching) scenario discussed in this paper was validated with the workload running on NVIDIA DGX-2 systems, however is applicable for the latest NVIDIA DGX A100 systems as well.

The target audience for this document includes

- Data scientists who are using or consider using AI models to derive value from data
- Solution architects: who need to think about how to integrate AI frameworks into their existing infrastructure and how to scale to serve an organization
- C-Suite: Create new business opportunities, innovate and generate data-driven insights using data

Use Case: Smoke Detection in Video Imagery for Prescribed Fire Management

Fire is a regular occurrence in forested landscapes throughout the world. Wildfires are a significant hazard to ecological systems around the world and can also be a threat to human safety. According to the California Department of Forestry and Fire Protection in the United States, "California experienced the deadliest and most destructive wildfires in its history in 2017 and 2018. Fueled by drought, an unprecedent buildup of dry vegetation and extreme winds, the size and intensity of these wildfires caused the loss of more than 100 lives, destroyed thousands of homes and exposed millions of urban and rural Californians to unhealthy air."[1]

Traditional methods of fire detection depended on fire lookout towers located on high ground with good visibility, where people watch for signs of fire or smoke. With the development of unmanned aerial vehicles (UAV) technology, advances in remote sensing, and information processing technology, it has become feasible to process and analyze images in real-time to detect smoke as an indicator of fire by analyzing aerial images of large forest areas. The use of UAVs and AI enhanced visual detection has significant potential to reduce damage. Deliberately set and managed prescribed fires may be one answer to reducing the spread and impact of wildfires. The mission of the National Prescribed Fire Training Center (PFTC) at Tall Timbers Research Station and Land Conservancy in Tallahassee, FL is "to maintain a center of excellence for prescribed fire, with an emphasis on actual field experience, in order to increase skills and knowledge and build confidence in the application of prescribed fire."[2]

During the Women-in-Fire Prescribed Fire Training Exchange (WTREX) from March 18-29, 2019, data scientists, UAV pilots, solution architects and videographers were invited to Tall Timbers Research Station to explore the growing role of women in fire management, while conducting prescribed fire operations designed to advance their formal qualifications in wildland fire management. These firefighters studied the effects wildland fire, communications and outreach, prescribed fire policy and planning, and—relevant to this use case— how AI, using DL neural networks and inferencing could be utilized to monitor smoke and wildfires. With the right hardware and software tools, firefighters could be better armed to fight fires with intelligent solutions before they rage out of control.

The objective of this use case was to simulate data flows from source to processing and apply AI techniques to detect smoke in video imagery to support prescribed fire management efforts. NVIDIA and IBM chose this use case to demonstrate how a well-designed AI Data Pipeline on the right infrastructure can enhance the ability to manage diverse data sets, simplify data processing, and enhance collaboration between experts. By streamlining AI model training, we hope to enable data scientists and wildfire experts to identify smoke at fire lines and inform wildfire management through prescriptive fire burns. The ultimate goal is to deploy AI models on or connected to unmanned aerial vehicles (UAV) that can send real time alerts to fire fighters at control burnout lines. Datasets used in this experiment are a combination of video data and imagery captured in the field and some existing reference datasets.

Intelligent AI systems using real-time imaging, weather, and topography, historical data may one day augment the knowledge and experience of firefighters and inform their use of prescribed fire as a means of reducing the threat of wildfire.

[1] California Department of Forestry & Fire Prevention: 2019 Fire Season (https://www.fire.ca.gov/incidents/2019/)
[2] Tall Timbers Research Station & Land Conservancy: eNews – PFTC/Tall Timbers Impact on Prescribed Fire around the World (https://talltimbers.org/pftc-tall-timbers-impact-on-prescribed-fire-around-the-world/)

What is the Purpose of an AI Data Pipeline?

An AI Data Pipeline should provide the systems, software, and efficient and repeatable processes for data science—simplifying the process and lower the cost of collecting data from ingest, classifying and preparing data, training ML/DL models, inference, and refinement. The specific architecture of an AI Data Pipeline will vary depending on the sources and types of data employed, and the steps required to process it. For these reasons, having flexible infrastructure is important to ensure efficiency and success.

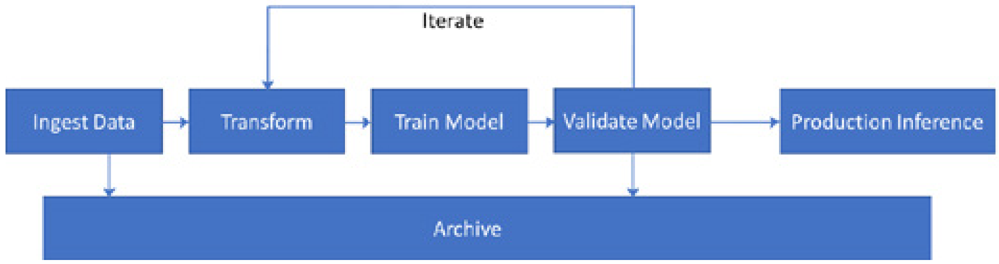


Figure 1*: Stages of a deep learning AI Data Pipeline*

Figure 1 shows the stages of the AI Data Pipeline for this wildfire management project. Details about each stage will be provided in other sections of this blueprint.

- **Ingest Data:** Data ingest often occurs at the edge but needs to be centrally available. In the DL AI Data Pipeline for the use case of smoke detection in video imagery, real-time video and still imagery are captured using UAV platforms and stored in IBM Cloud™ Object Storage (COS). A data scientist creates policies to classify, tag, and annotate the incoming data using IBM Spectrum Discover.
- **Transform:** Data transformation includes preprocessing operations to prepare data to be used by the DL algorithms. For images, this often includes file parsing, JPEG decoding, cropping, resizing, rotation, and color adjustment. Transformations can be performed on the entire dataset ahead of time, storing the transformed data for subsequent use. Many transformations can also be applied inline in the training pipeline, avoiding the need to store the intermediate data. In the use case of smoke detection in video imagery, the individual frames of video data were converted to images for further processing.
- **Train Model:** DL models are trained using neural networks. A neural network takes inputs which are processed using weighting factors specified by the data scientist. The model then produces a weighted prediction. Weighting factors are subsequently adjusted to yield more accurate predictions in subsequent iterations of training. For this use case, after having separated out a training set of labeled images, data scientists create a model with various parameters that cover image processing to identify smoke as opposed to other similar images.
- **Validate Model:** Once the model training stage completes with satisfactory accuracy, data scientists will measure the model accuracy against validation data — data that the model training process has not seen before. Our trained smoke model is run against a different but known data set to make inferences from the validation data and comparing the results with the labeled data.

- **Production Inference:** The trained and validated model is deployed to a system, for example an UVA, that can perform real-time inference. It will accept a single image as an input and output the predicted class—in our use case, the image data labelled as either containing smoke or not containing smoke.
- **Archive:** Data used in each training iteration may be saved indefinitely. Many AI teams archive data to object storage for cost effectiveness, ease of access, and simplicity.

All of these stages are important, but the transitions between stages are often just as crucial. Data must flow efficiently from ingest through preparation and then on to training and validation.

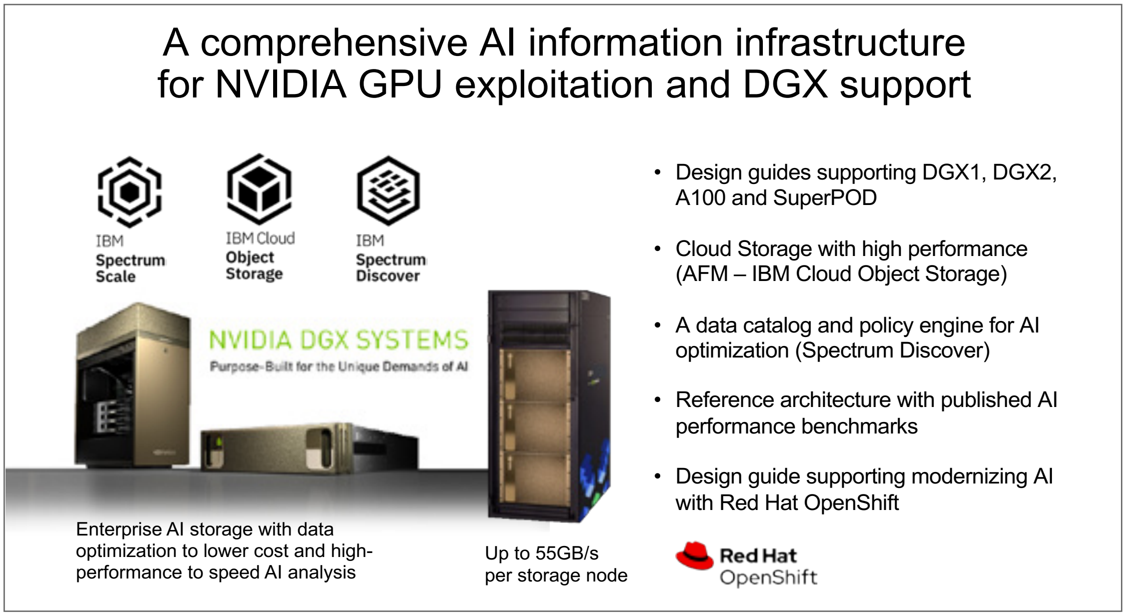


***Figure 2**: Faster results and more data with an AI information architecture*

Figure 2 illustrates the AI Data Pipeline architecture for our smoke detection use case using IBM Spectrum Storage for AI with DGX-2 systems. The ingested wildfire video and image data is stored in COS and then tagged and annotated by Spectrum Discover policies set up in advance by the data scientist. The data scientist queries IBM Spectrum Discover to select a dataset of interest and then submits the workload to IBM Spectrum Discover along with the dataset. IBM Spectrum Discover initiates the workload by scheduling it with IBM® Spectrum LSF, the workload management and job scheduler. This job submission triggers the dataset prefetch for the required data on the IBM Elastic Storage System 3000 powered by IBM Spectrum Scale™.

The smoke detection AI model[3] employed in this use case was developed by the Microsoft AI for Good team based on the MobileNet-V2 neural network vision model. IBM LSF schedules the model training job on the DGX-2 system and ESS 3000, producing the resulting trained model which then gets pushed back to COS. IBM Spectrum Discover then tags the new features and annotations and tracks model input, iterations, input parameters and output. The data scientist contrasts iterations for future forensics during inference.

Finally, the model is ready to be used for inference and data curation policies that are applied with IBM Spectrum Discover. Smoke is detected in the incoming dataset using the trained smoke detection model. The training process is iterative until data scientists are satisfied with the model.

[3] Code for the AI model can be found in Additional Resources at the end of this document.

Data Lake Ingest Powered by IBM Cloud Object™ Storage (COS)

A data lake is a central data repository that can hold vast amounts of unstructured data in its native format until the value has been discovered using other data sources, experimentation, and exploration through collaboration.

Using COS, organizations can build a centralized data repository, leveraging cost-effective and scalable storage that makes it possible to collect and store virtually unlimited amounts of data of any type and from any source. Data remains in its native format and doesn't need to be moved in and out of COS; rather, the COS based data lake is the persistent data store for the AI Data Pipeline. With inherent support for multi-site deployments and scalable performance, it provides a platform for global data ingest and multi-site data sharing.

Automatic Data Tagging with IBM Spectrum Discover

"Data scientists spend 60% of their time on cleaning and organizing data. Collecting data sets comes second at 19% of their time, meaning data scientists spend around 80% of their time on preparing and managing data for analysis". [4]

IBM Spectrum Discover automatically captures system metadata from source storage systems, enables the creation of custom metadata from search results, and the extraction of keyword metadata from file headers and content using content-based search capabilities and a rich API that enables users to extend and customize its capabilities.

The result is a rich layer of file and object metadata that is managed using one, centralized solution. Built-in connectors provide integration with heterogeneous storage system on premises and in the cloud, including COS, IBM Spectrum Scale, IBM Spectrum Protect™. backups, public cloud storage and several third-party storage systems. Metadata indexing enables rapid data queries and enables users to quickly locate data and create custom data sets for AI and analytics.

[4] Forbes: "Cleaning Big Data: Most Time-Consuming, Least Enjoyable Data Science Task, Survey Says", Gil Press, March 23, 2016 https://www.forbes.com/sites/gilpress/2016/03/23/data-preparation-most-time-consuming-least-enjoyable-data-science-task-survey-says/#3d63f5ad6f63

Data Accelerator for AI and Analytics with Elastic Storage® System 3000 (IBM ESS 3000)

(Deliver Data to the Training Platform)

Data science productivity and compute utilization requires the right data to be readily available for faster transformation, model training, inference or real-time analytics without the overhead of replicating data. In this blueprint, COS acts as a persistent cache that pre-fetches in real-time only required data from the capacity tiers, avoiding an extra copy of the data and helping to maintain a single source of truth. Model training is accelerated by prefetching the selected dataset in real-time into the Elastic Storage System 3000 DL environment from the COS data lake. This can be applied for either centralized training, or in a near-edge platforms for edge computing.

The ESS 3000, powered by IBM Spectrum Scale, provides high-speed data access for training, validation, and inference. A small all-flash storage platform can also be used to cache or manage data from NAS filers, Hadoop/Spark, IBM Tape, or multiple IBM Spectrum Scale clusters. Caching technology built into IBM Spectrum Scale is used in combination with IBM Spectrum Discover metadata functions such as tagging and classification in order to automate the selection of the right dataset into the persistent cache.

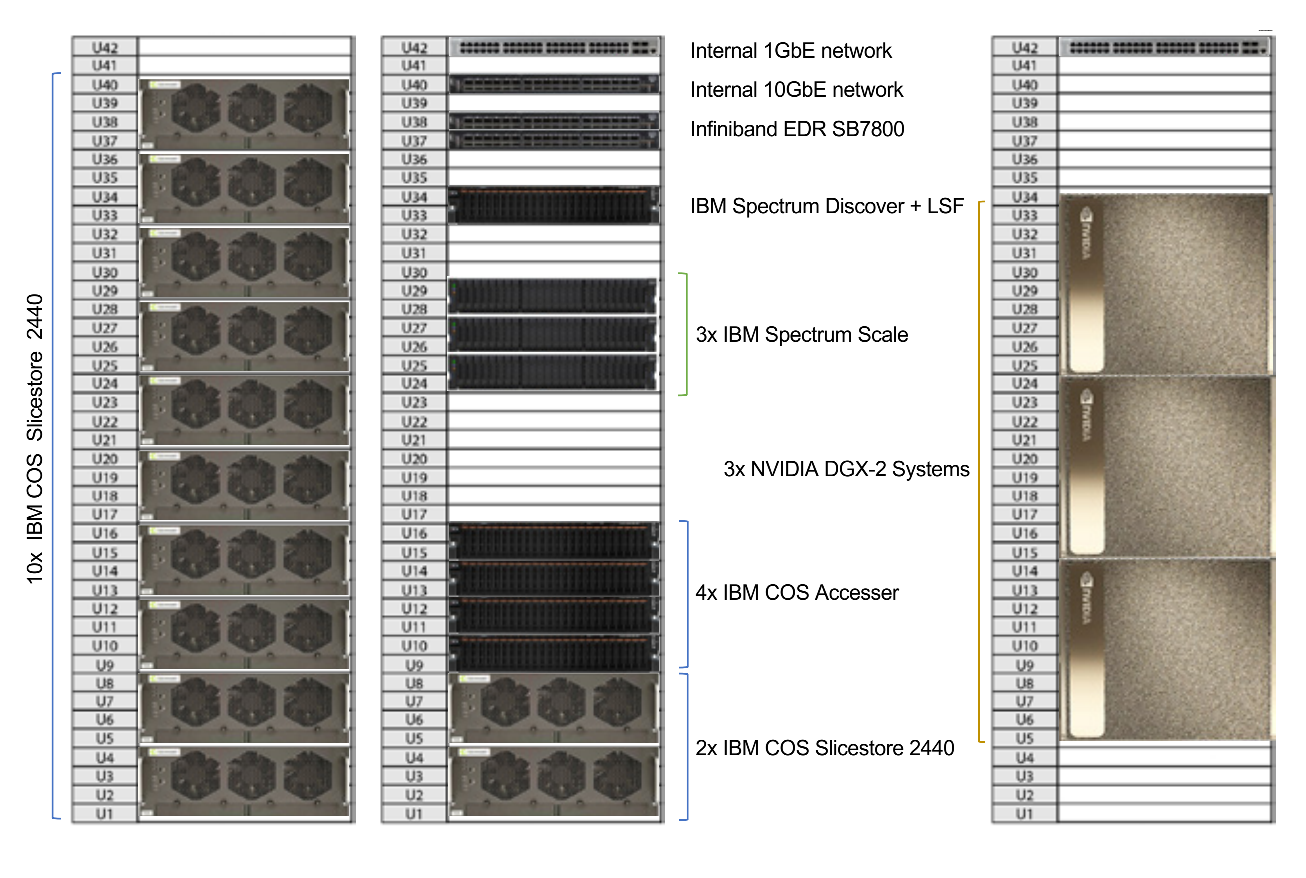
Training Cluster with NVIDIA DGX Systems

The training cluster for wild fire smoke detection use case is powered by DGX-2 systems. The DGX-2 system gives 10X compared to traditional server rich DL performance and is the world's first 2 petaFLOPS system with 16 interconnected GPUs in one chassis. Powered by NVIDIA CUDA-X™ software and the scalable architecture of NVIDIA® NVSwitch™ technology, the DGX-2 system is an AI platform for the most complex AI challenges.

Simplicity of Deployment and Scalability

The IBM Spectrum Storage for AI with DGX-2 systems provides scalability for capacity and performance. Deployments can start as small as a single ESS 3000, a single DGX-2 system attached to an InfiniBand network and expansions can be tailored to the increasing workload demands to hundreds of storage and compute systems. Additional DGX-2 systems can access all data with the IBM Software-Defined Storage solution, which gives data scientists and solution architects a simple and cost-effective growth path.

Overview of AI Data Pipeline Solution Architecture



Architecture

Software

- IBM Spectrum Discover (v2.0.1)
- IBM Cloud Object Storage Appliance software (v3.14.3)
- IBM Spectrum LSF (v10.1.0)
- IBM Spectrum Scale RAID (v5.3)
- Ubuntu 18.04
- Docker 19.04-py3
- NVIDIA AI OS
- NVIDIA GPU Cloud (container registry)
- TensorFlow 1.13.1
- Precision: FP16

Hardware

- IBM Spectrum Scale NVMe all-flash storage appliance
- Elastic Storage System 3000
- IBM Cloud Object Storage Accesser® and Slicestor® appliance
- Mellanox SB7800 Infini Band
- NVIDIA DGX-2 System

Network

- EDR InifiBand
- 10GB Ethernet *(management)*

Figure 3*: Physical architecture*

Figure 3 illustrates key components used with state-of-the-art software and supporting storage, network and compute technologies for each stage of the wild fire smoke detection AI Data Pipeline workflow.

Ingest

IBM Cloud Object Storage is deployed as a capacity tier for data ingest in the wild fire smoke detection DL end-to-end AI Data Pipeline. IBM COS solution consists of four Accesser and twelve Slicestor-2440 nodes installed and configured. The management node is running in a virtual machine.

The Accesser host software is responsible for encrypting/encoding data on ingest, decoding/ decrypting it when read, and managing the dispersal of data slices across a set of COS Slicestor nodes. The Slicestor node host software is responsible for the storage of slices.

The network between Slicestor and Accesser nodes as well as from Accesser to client nodes employs 10GbE links and uses Link Aggregation Control Protocol (LACP) bundling two links to increase bandwidth and redundancy for the COS internal and client network.

Transform

In the wild fire smoke detection AI DL Data Pipeline, IBM Spectrum Discover is deployed and configured in a virtual machine. IBM Spectrum LSF workload management and job scheduler is installed and configured in the host server. It provides a comprehensive set of intelligent, policy-driven scheduling. IBM Spectrum LSF clusters have one or more server hosts (also known as compute hosts) to run submitted jobs.

Ingesting COS event records into IBM Spectrum Discover requires the user to enable the Notification service on COS. Thereafter, the user must connect the COS system to the COS connector Apache Kafka topic on the IBM Spectrum Discover cluster.

The combination of Simple Authentication and Security Layer (SASL) and Transport Layer Security (TLS) is used to authenticate and encrypt the connection between the COS source system and the Apache Kafka brokers which reside in IBM Spectrum Discover.

Analyze /Train

To handle capacity and performance a total of three ESS 3000 in a single IBM Spectrum Sale cluster were deployed in this blueprint. Each ESS 3000 provides a pair of fully redundant storage servers within the IBM Spectrum Scale cluster. The ESS 3000s are connected over EDR InfiniBand with eight links to Mellanox® SB7800 switches to provide 200 GB/s for each ESS 3000. The DGX-2 systems also connect with eight links to the EDR InfiniBand switches. In addition to the high speed EDR InfiniBand fabric, the ESS 3000 and DGX-2 systems are connected to 10GbE networks for data exchange with COS as well as for system management.

Optimizing the Wildfire AI Data Pipeline

Accelerate Data Labeling Using IBM Spectrum Discover

One of the most powerful aspects of IBM Spectrum Discover is its ability to accommodate custom metadata created by users of the associated data or the Spectrum Discover system itself. Metadata can be collected simply by logically combining existing metadata, by leveraging header extraction tools such as Apache Tika (which is integrated into Spectrum Discover), or by creating software agents to extract metadata unique to the business.

There are three basic categories of user-defined metadata collection policies: AUTOTAG, CONTENT SEARCH, and DEEP-INSPECT. Tags are a prerequisite for policies because the purpose of a policy is to assign value(s) to one or more tags. A tag is a custom metadata field, or key:value pair that can be used to supplement system metadata that is automatically indexed by Spectrum Discover with additional organization-specific or domain-specific metadata.

Policies have a few common attributes such as the name of the policy, the filter associated with the policy, and one or more tags that the policy will assign values to. When defining a filter, use the same syntax as the WHERE clause in a standard SQL query. Ultimately the filter identifies a subset of objects to which the policy applies.

The first step for any policy is to create it. To do so, select the Metadata icon in the left pane of the GUI, select the Policies tab in the right (main) pane and select the Add Policy box as shown in Figure 4.

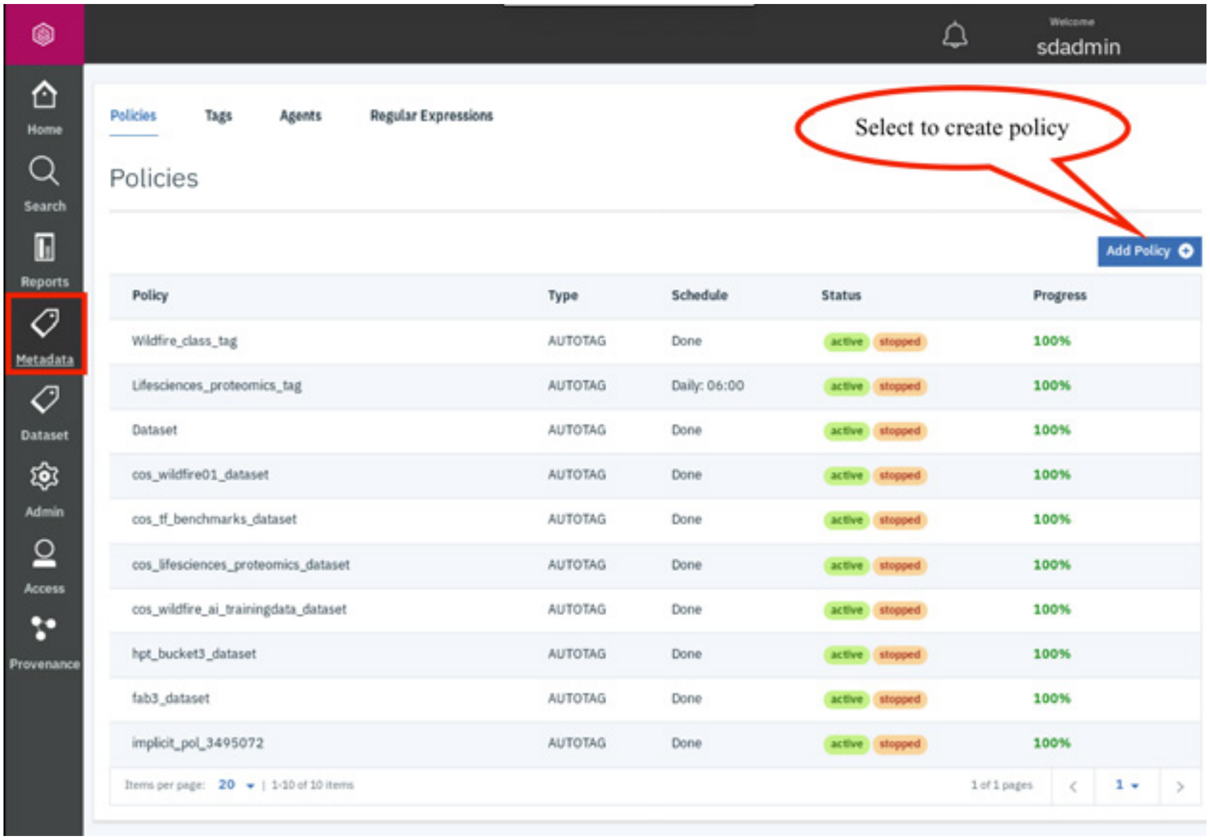


Figure 4: *Adding a new policy in IBM Spectrum Discover*

Autotag

For the smoke detection DL use case, we have defined the restricted tag **wildfire_class_tag** to have only values of **TRUE** or **FALSE.** Select the **Metadata** icon in the left pane of the Spectrum Discover GUI and select **Policies tab → Add Policy**. In the resulting **Add new policy** window, proceed as follows:

1. Give the policy a name.
2. Select **AUTOTAG** as the **Policy Type**.
3. Define the criteria for assigning the tag the desired value, i.e., define the Filter for the policy.
4. Select **Add tag**.
5. Select the desired tag using the **Tag** pull-down list.
6. Select the desired value using the **Values** pull-down list.

The **AUTOTAG** policy is very basic in its operation. From a logical perspective, it can be represented simply as:

if (<filter>) then <tag> = <value>

> **Note:**
>
> Data Source Connections are needed to connect IBM Spectrum Discover to the systems for scanning, indexing and auto tagging. Currently the supported types are IBM Cloud Object Storage, IBM Spectrum Scale, IBM Spectrum Protect, Network File System (NFS), and Amazon Simple Storage Service (S3).

Efficient Data Caching to High Performance Training Tier Using Data Accelerator for AI & Analytics

In the smoke detection DL use case, IBM Elastic Storage System 3000 provides the high-performance training tier. IBM COS is used as the capacity tier for data ingest from edge sources such as Unmanned Aerial Vehicles (UAVs). IBM Spectrum Scale caches the data and metadata with cache consistency. It maps each file with an object in IBM COS. The data preload first loads metadata to create a directory structure. All files will be preloaded and appear in the newly created directories before the data arrives. The preload loads the data in parallel streams to IBM Spectrum Scale. After that early access to preloaded metadata only file will instantly load the data for that file as shown on Figure 5.

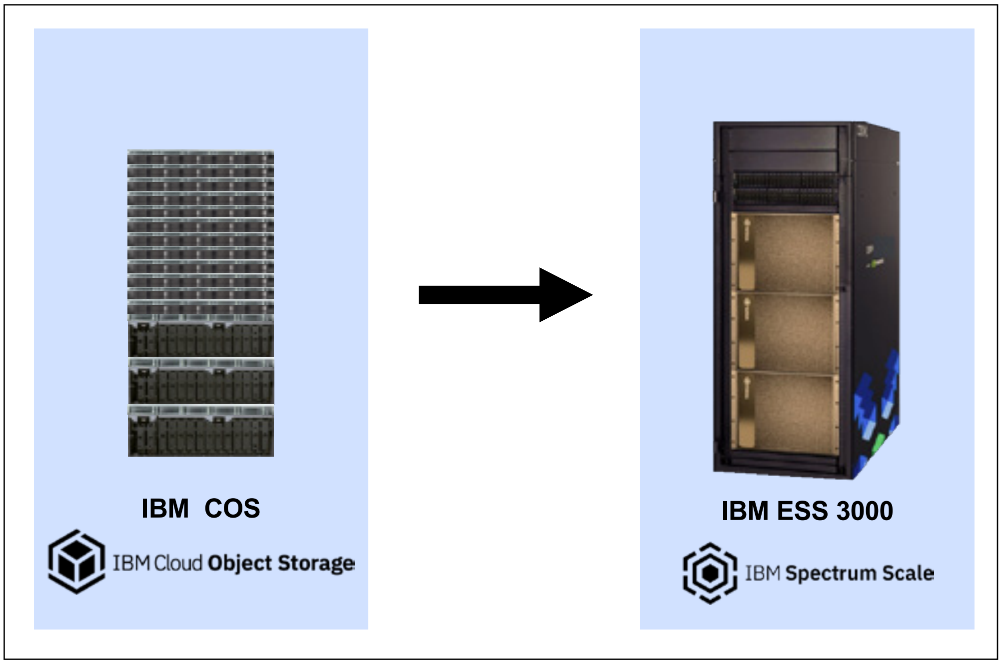


***Figure 5**: High speed performance tier*

All durable data and metadata reside in IBM COS. IBM Spectrum Scale only caches the data and metadata required for model training purposes.

> **Note:**
> IBM Spectrum Scale can also be used to cache data from any cost-optimized capacity tiers like IBM COS and data lakes built using NAS filers or IBM Spectrum Scale itself.

In the smoke detection DL use case, IBM Spectrum Discover is used to enable metadata management functions such as tagging and classification in order to automate selection of the right dataset for prefetching into the persistent cache. See Figure 6.

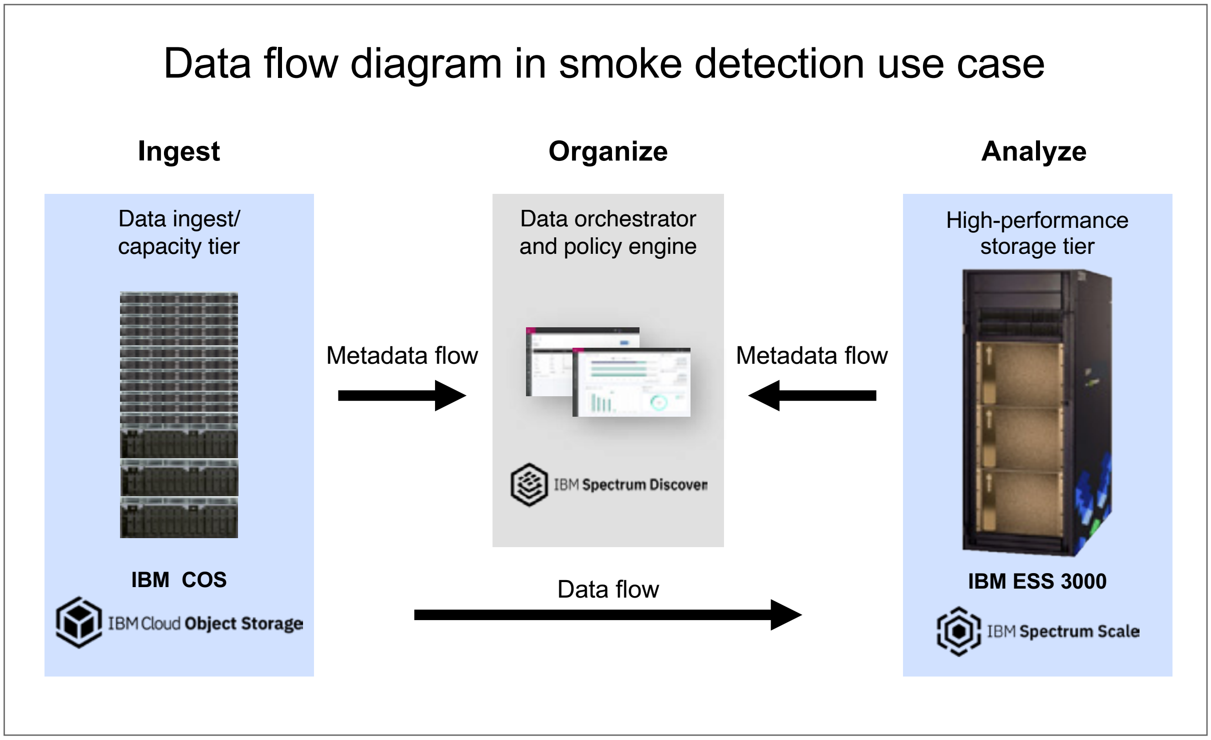


***Figure 6**: Data Flow*

Smoke Detection AI Model

Using the smoke detection AI model, smoke is identified by examining image and video files produced by UAVs. The smoke detection AI model is developed using the MobileNet-V2 family of general-purpose computer vision neural networks designed with mobile devices in mind to support classification, detection and more for TensorFlow.

MobileNet-V2 provides the ability to run deep neural networks on personal mobile devices improving user experience, offering anytime/anywhere access with benefits for security, privacy, and energy consumption. As new applications emerge allowing users to interact in real time, so does the need for ever more efficient neural networks. MobileNet-V2 is the state of the art for mobile visual recognition including classification, object detection and semantic segmentation.

MobileNet-V2 builds upon the ideas using depth-wise separable convolution as efficient building blocks and introduces two new features: linear bottlenecks between the layers, and shortcut connections between the bottlenecks. The bottlenecks encode the model's intermediate inputs and outputs while the inner layer encapsulates the model's ability to transform from lower-level concepts such as pixels to higher level descriptors such as image categories. Finally, as with traditional residual connections, shortcuts enable faster training and better accuracy.

A neural network consists of an input layer and an output layer, separated by a number of hidden layers as shown on Figure 7.

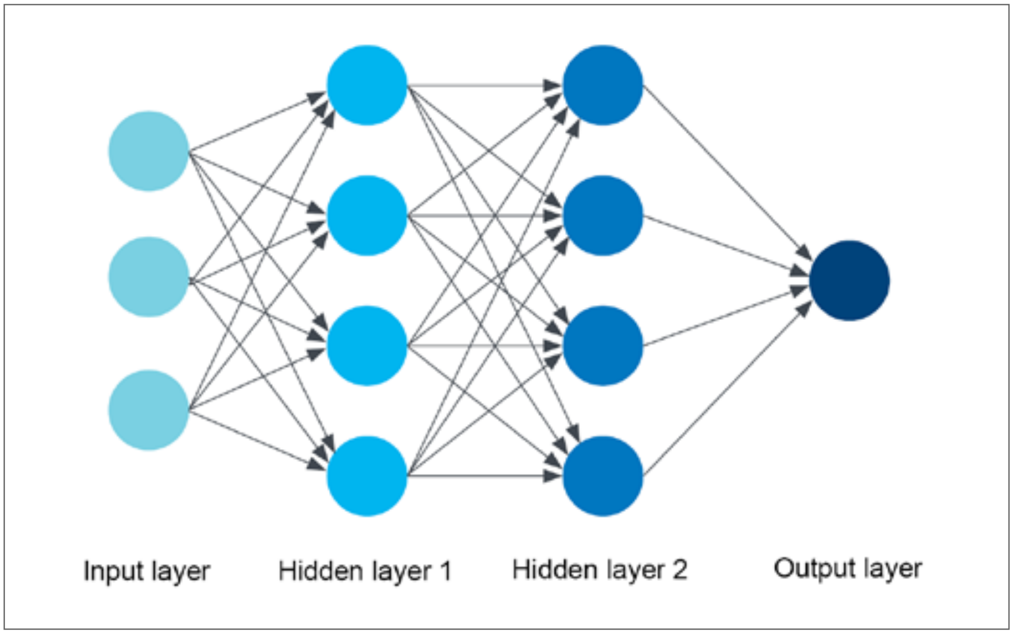


***Figure 7**: Neural Network Architecture*

The nodes in the hidden layers of a neural network mostly execute simple mathematical calculations, called activation functions. The way the nodes are interconnected, the mathematical functions they use, and the breadth and width of the network all contribute to the architecture of the deep neural network and determine the type of network available. Smoke Detection using Object Detection.

Smoke Detection Using Object Detection

Object detection is a computer vision technique for locating instances of objects in images or videos. Object detection algorithms typically leverage DL to produce meaningful results. When humans look at images or video, we can recognize and locate objects of interest within a matter of moments. The goal of object detection is to replicate this intelligence using a computer.

In the smoke detection DL use case, the first step is training a model to detect the objects, then the trained model can be loaded for inference.

> **Note:**
>
> Code snippets can be found in the provided source code in lines 42-48 where the object detection procedure is called, and in lines 168-244 where the object detection itself is implemented. [1]
>
> All credit for this AI model goes to Anusua Trivedi, Senior Data Scientist for Microsoft AI for Good Team. The AI model is Copyright (c) Microsoft Corporation. All rights reserved.

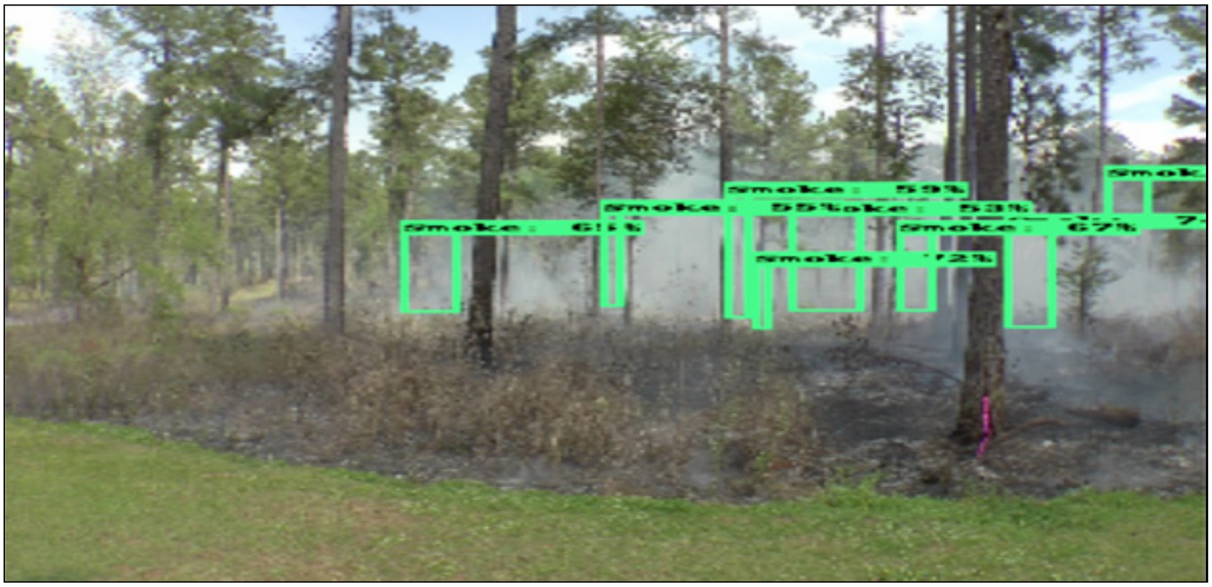


***Figure 8**: Output of smoke detection model using object detection*

Figure 8 shows the result of smoke detection in image data using the AI model and the object detection method.

Smoke Detection Using Image Segmentation

Image segmentation is the process of partitioning a digital image into multiple segments (sets of pixels, also known as super-pixels). The goal of image segmentation is to cluster pixels into salient image regions, i.e., regions corresponding to individual surfaces, objects, or natural parts of objects. Image segmentation is typically used to locate objects and boundaries (lines, curves, etc.) in images.

More precisely, image segmentation is the process of assigning a label to detected areas in an image such that pixels with the same label share certain characteristics. The result of image segmentation is a set of segments that collectively cover the entire image, or a set of contours extracted from the image. See Figure 9.

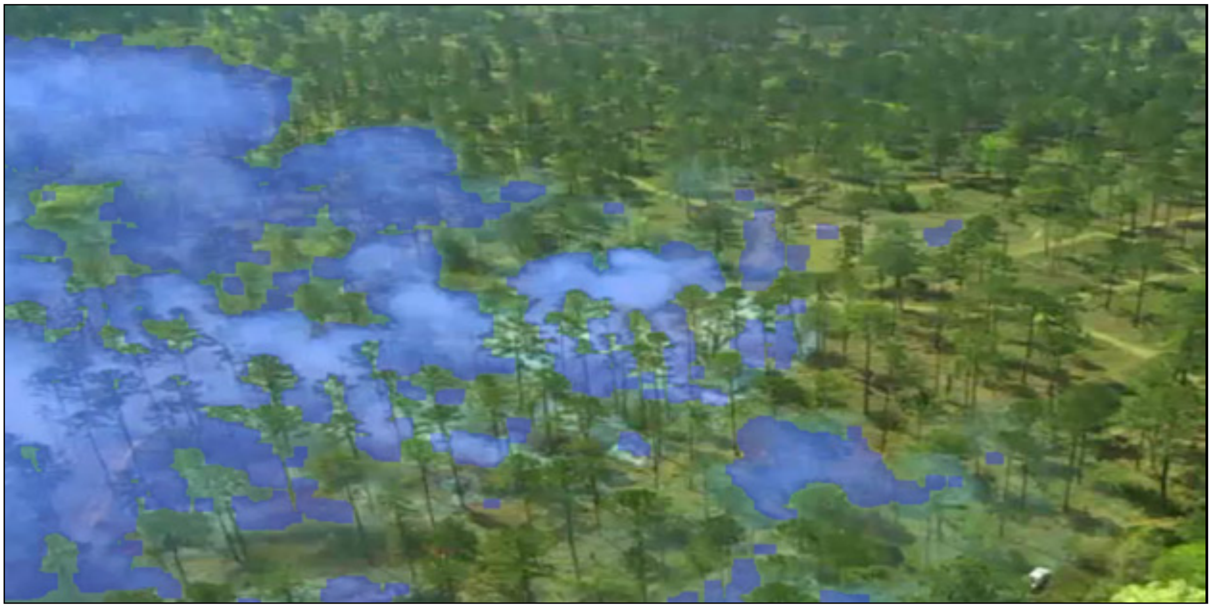


Figure 9: Result of smoke detection using image segmentation. The blue overlay represents regions of dense smoke

Note:

For additional information on the image segmentation algorithm refer to end note. [2]

Imagery used to create dataset for smoke segmentation was provided by AeroVironment, Inc.

Conclusion:
Take Advantage of End-to-End AI Data Pipeline Efficiency

This blueprint provides guideline to help plan and implementation a complete AI Data Pipeline. It outlines the essential software and supporting systems to build an AI Data Pipeline for efficiently collecting, preparing, and managing all the data to train, validate, and operationalize an AI algorithm.

We have used smoke detection as a use case to demonstrate how to implement an AI Data Pipeline, but this example can be extended to any industry or use case from autonomous driving to fraud detection.

We demonstrated how to effectively use the following IBM Storage offerings for data and AI use cases to implement an AI Data Pipeline while leveraging high performing NVIDIA DGX-2 systems as AI server technology.

- **IBM Elastic Storage System 3000 as the high-performance tier**
- **IBM Cloud Object Storage as the data lake / capacity tier**
- **IBM Spectrum Discover as the metadata management platform**
- **IBM Data Accelerator for AI & Analytics solution to automatically prefetch required data from capacity tier to high-performance tier**

Appendix 1: Components of the AI Data Pipeline

NVIDIA DGX-2 System

Increasingly complex AI demands unprecedented levels of compute. The NVIDIA DGX-2 system (see Figure 10) is the world's first 2 petaFLOPS system, packing the power of 16 of NVIDIA advanced GPUs to support the latest DL model types.

Each DGX-2 system integrates sixteen NVIDIA Tesla™ V100 Tensor Core GPUs in a NVIDIA NVSwitch fabric using NVIDIA NVLink™ technology. This high bandwidth low-latency GPU-to-GPU fabric provides 2 petaFLOPS of DL computing capacity by eliminating bottlenecks and intermediary GPU hops, and enables scalable multi-GPU training while eliminating the PCIe-based interconnect bottleneck found in traditional architectures. The DGX-2 system includes eight Mellanox VPI cards enabling EDR InfiniBand or 100GbE network ports for multi-node clustering with high speed RDMA capability for exceptional storage-to-DGX system data rates.

Figure 10: *NVIDIA DGX-2 System*

The DGX-2 system is powered by the NVIDIA CUDA-X software stack which includes the DGX operating system and optimized DL containers engineered by NVIDIA for GPU-accelerated performance. This CUDA-X software stack facilitates rapid development and deployment on a one or multiple DGX-2 systems with multi-GPU and multi- system scale-up of applications saving time and developer effort. The software components offered with the NVIDIA DGX POD™ include cluster management and orchestration tools, and workload scheduling which can be leveraged for management of the AI workloads.

Mellanox InfiniBand Network

Mellanox is an industry-leading supplier of switches, cables, and network adapters for Ethernet and InfiniBand. Mellanox delivers a complete interconnect solution for high performance GPU clusters used for AI workloads, leveraging advanced technologies such as RDMA, GPUDirect® technology to accelerate GPU-to-GPU communications, SHARP™ technology to accelerate machine learning algorithms by providing state-of-the-art in-network computing capabilities.

The Mellanox® EDR InfiniBand network provides scalability of the GPU workloads and datasets as well as inter-node communication between DGX-2 systems.

IBM Elastic Storage System 3000 and IBM Spectrum Scale

***Figure 11**: IBM Elastic Storage System 3000*

The Elastic Storage System 3000 is an all-flash storage platform powered by IBM Spectrum Scale. It provides low-latency and a very high throughput in an easy to deploy solution. See Figure 11.

IBM Spectrum Scale is software-defined storage that scales out and is optimized for AI data workloads. IBM Spectrum Scale on Elastic Storage Systems provide the storage for two of the worlds faster supercomputers: Summit and Sierra, installed at Oak Ridge National Laboratory and Lawrence Livermore National Laboratory. Summit has the capacity of 30B files and 30B directories and can create files at a rate of over 2.6 million file input/output operations per second. That is equivalent to opening every book in the US Library of Congress in ten seconds.[5]

IBM Spectrum Scale is a distributed, parallel filesystem that scales in both capacity and performance. It provides a single data plane across multiple storage platforms and protocols, including NFS, SMB, Object, HDFS, and a POSIX interface. A single namespace allows data preparation, training, and inference applications to access data in place.

[5] The fastest storage for the fastest system: Summit (https://www.ibm.com/blogs/systems/fastest-storage-fastest-system-summit/)

IBM Spectrum Scale enables data to be tiered automatically and transparently across multiple storage platforms including NVMe Flash, hard disk drive (HDD), tape, and cloud to provide both cost effective capacity and high-performance data transfer that is transparent to the end users and automatically managed.

Unlike legacy NFS or SMB network storage, the file system can be accessed concurrently by all compute nodes in the configuration to efficiently meet the capacity and performance requirements of modern scale out applications such as AI.

The IBM Elastic Storage Server 3000 uses two redundant controllers and storage erasure code technologies to provide high levels of storage performance, availability, and reliability.

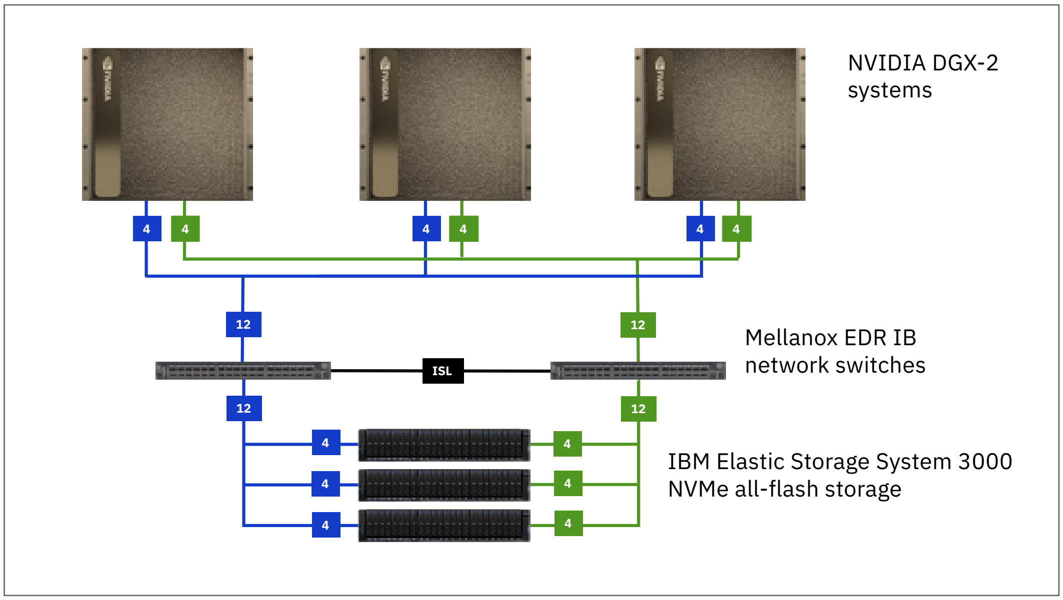


Figure 12: *Three NVIDIA DGX-2 Systems + Three ESS units*

Figure 12 depicts three DGX-2 systems with three 2U ESS 3000 units. This is the blueprint used in the IBM Elastic Storage Server 3000 with DGX-2 systems benchmark, however only two ESS 3000 units were necessary to saturate file system throughput of the DGX-2 systems and obtain the DGX-2 system model images/sec results reported in this paper. Regardless of the number of storage systems used, customers get shared storage and a simple and flexible growth path.

IBM Spectrum Discover

IBM Spectrum Discover is modern metadata management platform providing data insight for unstructured storage across multiple storage solutions and workloads. IBM Spectrum Discover connects to heterogeneous data sources including COS, IBM Spectrum Scale, IBM Spectrum Protect backup, public cloud, and several third-party storage systems with a common metadata index. Figure 13 shows IBM Spectrum Discover capabilities.

Scalable to billions of files and objects, it provides a rich metadata layer on top of these storage sources. The metadata can be customized and extended so data scientists can efficiently manage, classify and gain insights from massive amounts of unstructured data.

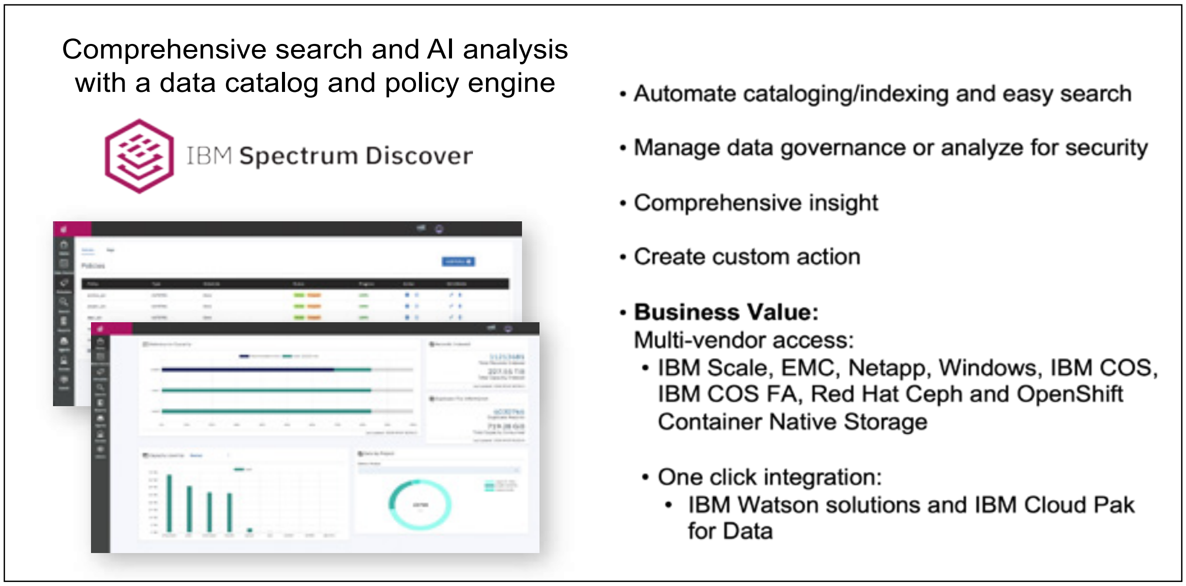


Figure 13: *IBM Spectrum Discover*

IBM Spectrum Discover can read metadata information from the source storage system and automatically catalogue the information in real-time with little to no impact on the underlying storage. IBM Spectrum Discover includes Apache Tika for file inspection and extensible APIs for custom workflows and data tagging/ This enables Spectrum Discover to deliver rapid results of complex queries, or multi-faceted searches about the data. Search results may also be visualized in a drill down GUI dashboard.

Spectrum Discover is extensible, which provides a mechanism for communication with applications that can provide even greater insight into selected data by interrogating the contents of the full data, rather than just the metadata. The Spectrum Discover platform embeds an Apache Kafka instance, which enables a communication stream that can publish and subscribe to streams of records, similar to an enterprise messaging system

To capitalize on the flexible, extensible architecture of IBM Spectrum Discover, two Application Programming Interfaces (API) are provided:

- Policy management API
- Action agent API

The action agent API is used to establish the Apache Kafka topic interfaces used for messaging, as well as carrying out work to be done on selected data, hence the name action agent.

The policy management API is a RESTful web service that is designed to create, list, update, and delete policies. The policy management API also provides the means to initiate a policy immediately or schedule it to run on a given schedule.

Business-oriented data mapping can be carried out by the policy management API. An example of business-oriented data mapping is adding a project name to the catalog based on the location, or path, to the data.

IBM Spectrum LSF

The IBM Spectrum LSF Suite redefines cluster virtualization and workload management by providing a tightly integrated solution for demanding, mission-critical HPC and AI environments that can increase both user productivity and hardware utilization while decreasing system management costs. The heterogeneous, highly scalable and available architecture provides support for traditional high-performance computing and high throughput workloads, as well as for big data, cognitive, GPU machine learning, and containerized workloads.

With a comprehensive set of intelligent, policy-driven scheduling features, IBM Spectrum LSF enables you to make the most of the compute infrastructure resources and help with application performance. The highly scalable and available architecture allows to schedule complex workloads and manage up to petaflop-scale resources

IBM Cloud Object Storage

IBM Cloud Object Storage (COS) is a breakthrough object storage platform that helps solve storage challenges for enterprises worldwide. It uses an innovative and cost-effective approach for storing large volumes of unstructured data while still ensuring scalability, security, availability, reliability, manageability and flexibility (See Figure 14). It also can be provisioned with Analytics Engine to store data from multiple sources and quickly gain insights.

COS consists of following key elements:

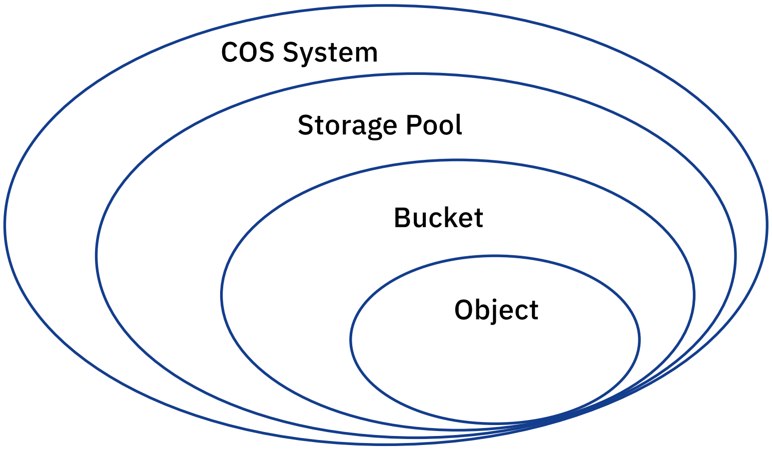


Figure 14: *Logical concepts in COS*

- **Bucket:** A bucket refers to vault (vault mode) or container (container mode) in a COS system. A bucket is a logical abstraction that is used to store the data.
- **Object:** An object refers to user data that is uploaded to a COS system. Typically, it is a file and the object metadata that is stored together with the file.
- **IBM COS Manager node:** A system component that provides a management interface that is used for administrative tasks such as system configuration, storage provisioning and monitoring the health and performance of the system. A Manager node (also referred as Manager) can be deployed as a physical appliance, VMware virtual machine or Docker container.
- **IBM Cloud Object Storage Accesser® node:** A system component that encrypts and encodes data on write or decodes and decrypts data on read. It is a stateless component that manages the transformation of the data and presents the storage interfaces to the client applications. An Accesser node can be deployed as a physical appliance, VMware virtual machine, Docker container or as an embedded Accesser on a Slicestor appliance.

- **IBM Cloud Object Storage Slicestor® node:** A system component that is responsible for storing the data. It receives data from the Accesser node on write and returns data to the Accesser node as required on read. Slicestor nodes are deployed as physical appliances.
- **Device set:** A device set is defined by a group of Slicestor devices. Device sets can be spread across one or multiple data centers.
- **Storage pool:** A storage pool is a logical grouping of one or more device sets which together provide the physical storage resources for one or more buckets.
- **Access pool:** An access pool is a logical grouping of one or more Accesser nodes that are used to access the data.

COS is integrated with IBM Analytics Engine, IBM Watson® Studio, IBM Cloud SQL Query and other IBM Cloud services to provide self-service data analytics and business intelligence solutions that go well beyond the scalability, security and cost efficiencies of traditional solutions.

Appendix 2:
Essentials of Designing and Building an AI Data Pipeline

Determining Data Requirements

It is critical to assess the data the AI Data Pipeline must support. Without the necessary dataset, data scientists will have to identify how to obtain the missing data. In some cases, the data scientist may have to table the use case until a process and infrastructure is in place to gather the necessary data.

For organizations new to big data and AI, IBM recommends that customers consider thinking about all of the data assets likely to be sourced and used over time. Having the right governance policies in place helps ensure that datasets are available, usable, and consistent when the data science teams need them. It is also important to ensure that key regulations for data protection and data privacy are met.

An important realization when thinking about data is that not all datasets can be treated equally; there are many different data types. A use case may require only one data type, or it could encompass multiple types. The data types used may determine the software tools needed in each stage of the AI Data Pipeline. IBM also recommends considering what the growth path looks like over the course of the expected project lifespan. Will the project scale to exabytes in a decade? If so, what does the growth path look like? What data will need to be kept? What strategies will be used for cold storage and archiving?

A few common data types along with examples and sources are shown in Table 1.

<table>
<tr><th>Data type</th><th>Examples</th><th>Sources</th></tr>
<tr><td>Image</td><td>• Assembly line cameras
• Medical imaging
• Geospatial Imagery
• Geologic images
• Thermal Imaging
• Lidar 3D Point Cloud</td><td rowspan="3">• NFS
• GPFS</td></tr>
<tr><td>Video</td><td>• Security cameras
• Autonomous vehicles
• Drones
• Cobots</td></tr>
<tr><td>Audio</td><td>• Voicemail
• Customer service calls
• Acoustic data</td></tr>
<tr><td>Time-Series Data</td><td>• Internet of Things (IoT)
• Securities prices
• Scientific data</td><td>• NoSQL databases (Cassandra, AeroSpike)</td></tr>
<tr><td>Text</td><td>• Log data
• Unstructured text
• Documents</td><td>• Splunk, ELK, etc.
• NFS, Hadoop/HDFS
• NoSQL databases (MongoDB)</td></tr>
<tr><td>Graph</td><td>• Social media data
• GPS navigation</td><td>• Graph Databases</td></tr>
</table>

Table 1: *Common data types in DL*

Planning Data for Training

The planning data stage for implementing an AI Data Pipeline encapsulates several functions. The essential purpose of planning data is to create a dataset that is suitable for training and validating a DL model. The planning or preparing of data isn't necessarily a discrete function. These steps can see some preprocessing during ingest and some prior to training, or it can all be done as a part of the training process.

Data planning generally can include:

- **Exploring the data.** What is the premise for the training model, and what features of the data are likely to be predictive?
- **Cleaning up data types and data formats.** Training will go more smoothly if the data is consistent; however, most won't want it to be more consistent than the live traffic the model will receive.
- **Adjusting the training dataset.** It is important to make sure the feature a model is being trained on is adequately represented. For example, an effective model for anomaly detection cannot be built well if the dataset consists only of images of "good" parts.
- **Labeling datasets.** For supervised learning, datasets need to be properly labeled.
- **Splitting the dataset into training, validation, and testing sets.** Data scientists need enough data to provide a training set, a separate validation set that is used during training but which the model is not trained on, and a testing set to assess the performance of the trained model.

Data planning and preparation, especially in early phases of model development, is frequently an iterative, exploratory process, aimed at understanding which processes deliver the best results. Table 2 shows common data preparation activities for various data types.

Data type	Examples
Image	• Make sure that all images are the same size and resolution. • Make sure that all images are black and white or all color. • Label features in images. • Correct any data imbalances (using over-sampling, under-sampling, data augmentation, class weights).
Video	• Extract JPEGs or BMPs of each frame. • Scale image size up or down as needed. • Correct any data imbalances (using over-sampling, under-sampling, data augmentation, class weights).
Audio	• Choose sampling rate. • Transform signal from time domain to frequency domain. • Use magnitude compression.
Time-Series Data	• Normalization (all values between 0 and 1). • Standardization (rescaling values so mean is 0 and standard deviation is 1).
Text	• Normalization (eliminate case and punctuation, convert numbers to text, etc.). • Tokenization (split text into “tokens” that represent words, sentences, or paragraphs). • Noise reduction (remove headers and footers, HTML, metadata, etc.).

Table 2: *Common data planning activities for various data types*

Acknowledgements

We gratefully acknowledge the contributions to this blueprint by our valued colleagues from NVIDIA, Jacci Cenci, Satinder Nijjar, and Anish Singhani. Special thanks to Anusua Trivedi, Senior Data Scientist for Microsoft AI for Good Team for providing the smoke detection AI model. We would also like to thank the team at Arrow Electronics. We could not have completed this study without help from our key IBM team members, Constantine Arnold from the IBM Research Lab for initial ideas, creating test system and establishing pipelines and benchmarks, and Brian Porter for support and guidance.

Our sincere appreciation and thanks to all these individuals, who provided insight and expertise that greatly assisted in the research for this paper.

Additional Resources

- **[1]** Microsoft Smoke detection model code snippet https://github.com/antriv/Smoke_Detection_AI_Model/blob/master/smoke_OD_inference.py
- **[2]** Real-Time Free space Segmentation on Autonomous Robots for Detection of Obstacles and Drop-Offs https://arxiv.org/pdf/1902.00842.pdf
- IBM Spectrum Discover: Metadata Management for Deep Insight of Unstructured Storage http://www.redbooks.ibm.com/abstracts/redp5550.html
- IBM Cloud Object Storage Concepts and Architecture http://www.redbooks.ibm.com/redpapers/pdfs/redp5537.pdf
- Forest Fire Detection Solution Based on UAV Aerial Data https://pdfs.semanticscholar.org/3f90/0c48bf78d5fdc2ace81c68663f9ee8ae131b.pdf
- Predicting Burned Areas of Forest Fires: An Artificial Intelligence Approach https://www.researchgate.net/publication/274889641
- Lean, Green, AI Machines: 5 Projects Using GPUs for a Better Planet https://blogs.nvidia.com/blog/2019/04/22/earth-day-ai/
- A Review of Deep Learning Methods and Applications for Unmanned Aerial Vehicles https://www.hindawi.com/journals/js/2017/3296874/
- Introduction Guide to the IBM Elastic Storage Server http://www.redbooks.ibm.com/redpapers/pdfs/redp5253.pdf
- NVIDIA DGX SuperPOD with IBM Elastic Storage Server delivers record performance https://devblogs.nvidia.com/dgx-superpod-world-record-supercomputing-enterprise/ https://www.nvidia.com/content/dam/en-zz/Solutions/data-center/gated-resources/nvidia-circe-reference-architecture.pdf

IBM Systems
3039 Cornwallis Road
RTP, NC 27709

Produced in the United States of America