A Scalable AI Data Pipeline for Storing and Processing Ingested Data

Sanhita Sarkar, Ph.D

Global Director,

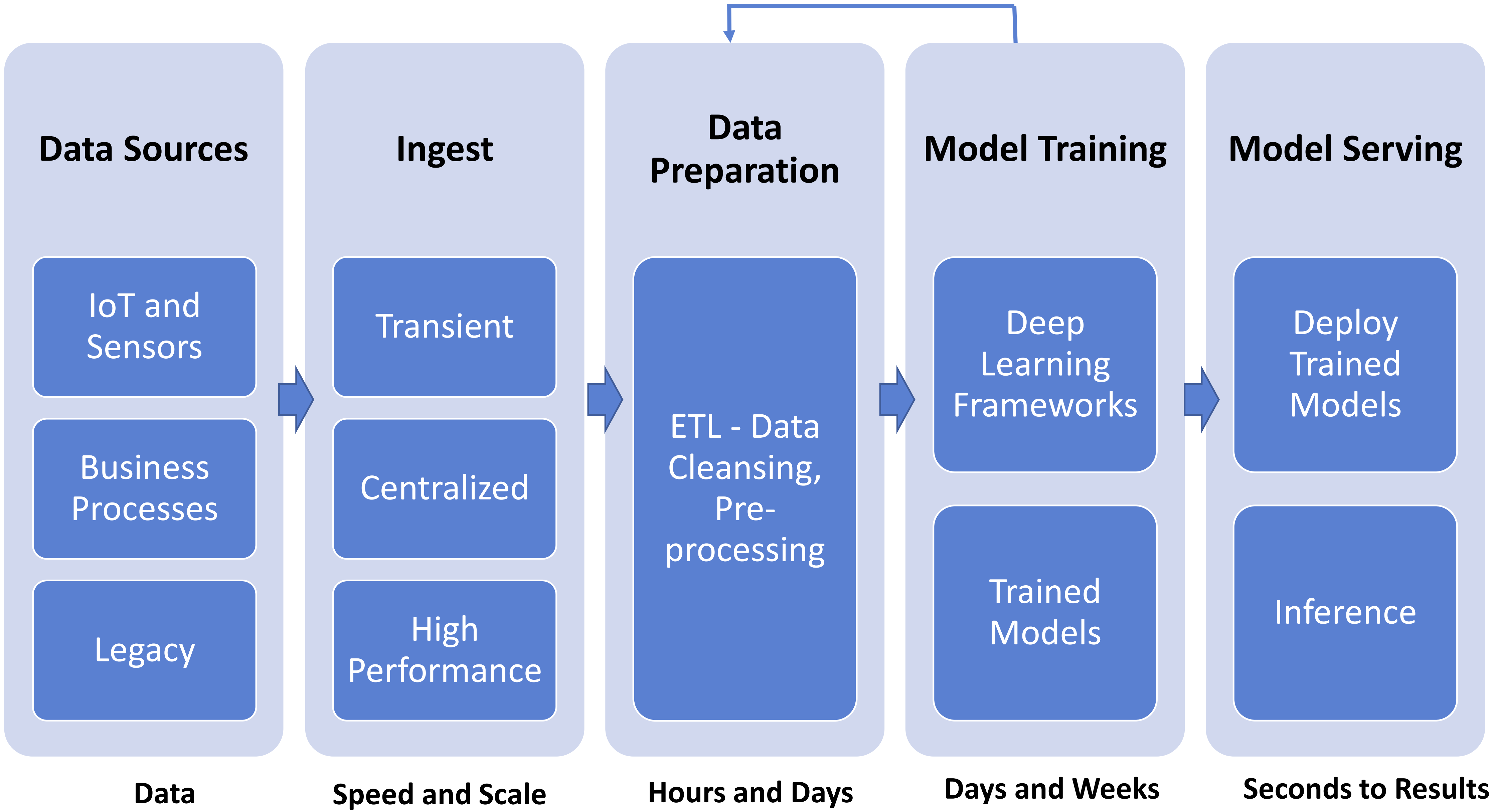
Analytics Software Development

August 8, 2019

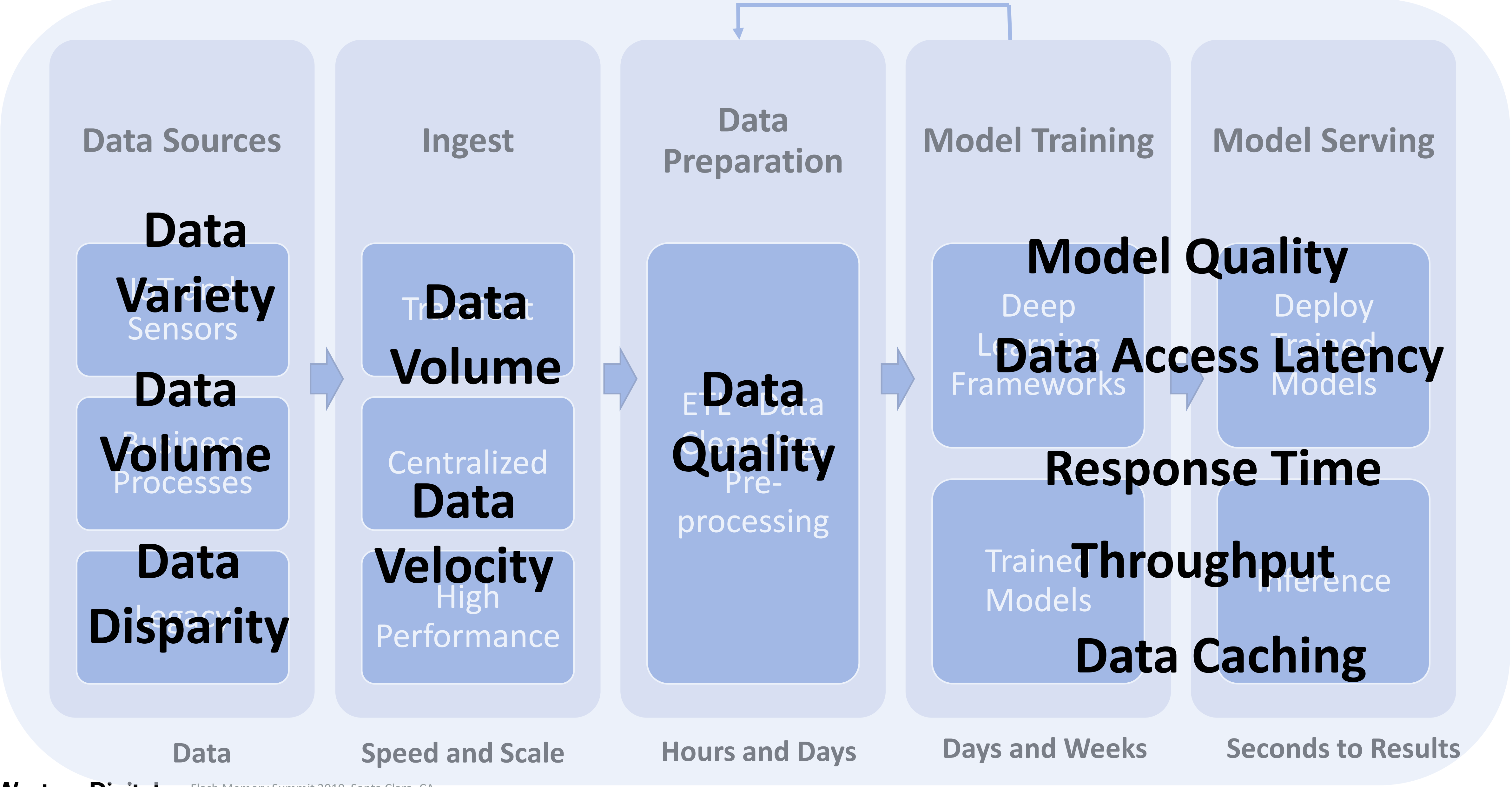
Topics Covered

1. An AI Data Pipeline - Stages and Requirements
2. Disaggregated Architectures for AI Implementations
3. Training and Inference Performance on a Disaggregated Architecture
4. Data Ingestion Performance with Apache Kafka© to an NVMe™ All-Flash Array
5. A Real-World Use Case
6. Summary and Best Practices

AI Pipeline is Data Intensive



AI Pipeline has Varying Characteristics and Performance Requirements



Flash Memory Summit 2019, Santa Clara, CA

AI Pipeline has Varying Infrastructure Requirements

Data Sources
- IoT and Sensors
- Business Processes
- Legacy

Data

Ingest
- Transient
  - *Throughput-oriented Temporary Tier (NVMe™/Object Storage)*
- Centralized
  - *Throughput-oriented, Globally Accessible Capacity Tier (Object Storage)*
- High Performance
  - *Throughput-oriented, High Performance Tier (NVMe™)*

Speed and Scale

Data Preparation
Throughput-oriented, Random I/O Performance and Capacity Tier (NVMe™/Hybrid)

ETL - Data Cleansing, Pre-processing

Hours and Days

Model Training
- Deep Learning Frameworks
- Trained Models

Days and Weeks

Model Serving
- Deploy Trained Models
- Inference

Seconds to Results

Low Latency, Throughput-oriented, Scalable, Random I/O Performance Tier (NVMe™)

Scalable, Large/Sequential I/O, Capacity Archival Tier (Object Storage)

Topics Covered

1 An AI Data Pipeline - Stages and Requirements

2 Disaggregated Architectures for AI Implementations

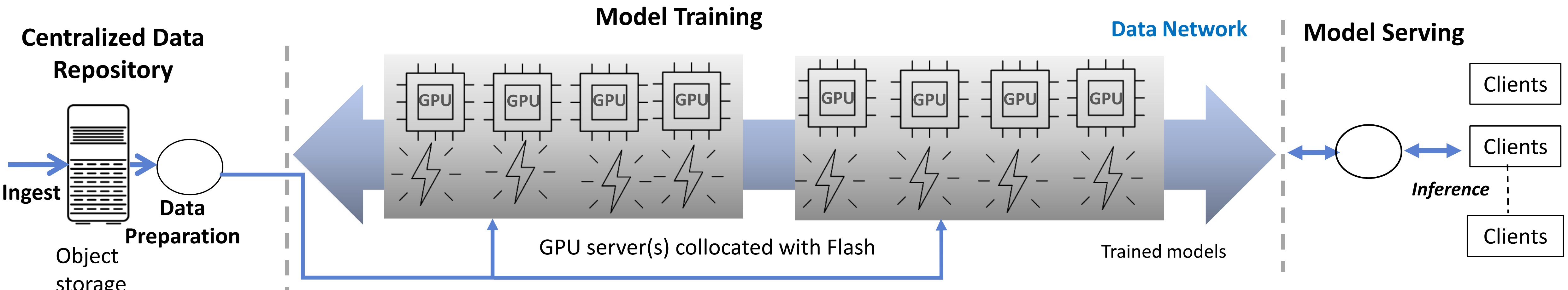
3 Training and Inference Performance on a Disaggregated Architecture

4 Data Ingestion Performance with Apache Kafka© to an NVMe™ All-Flash Array

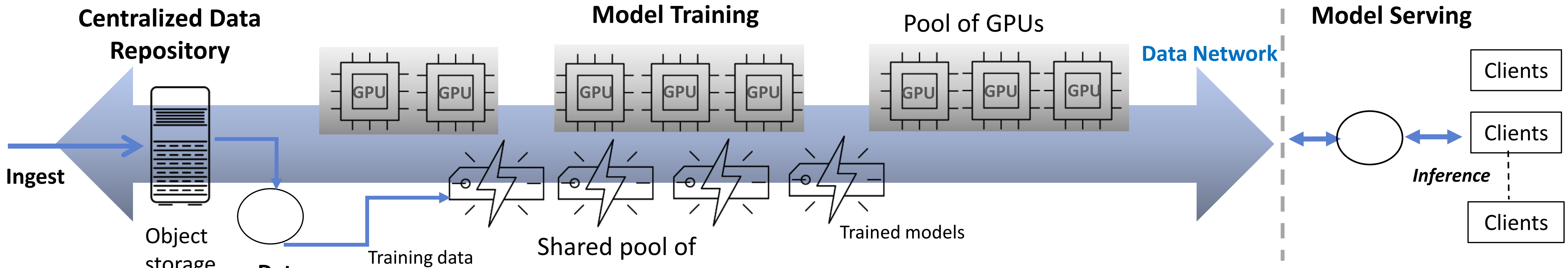
5 A Real-World Use Case

6 Summary and Best Practices

Aggregated vs. Disaggregated Architecture for AI

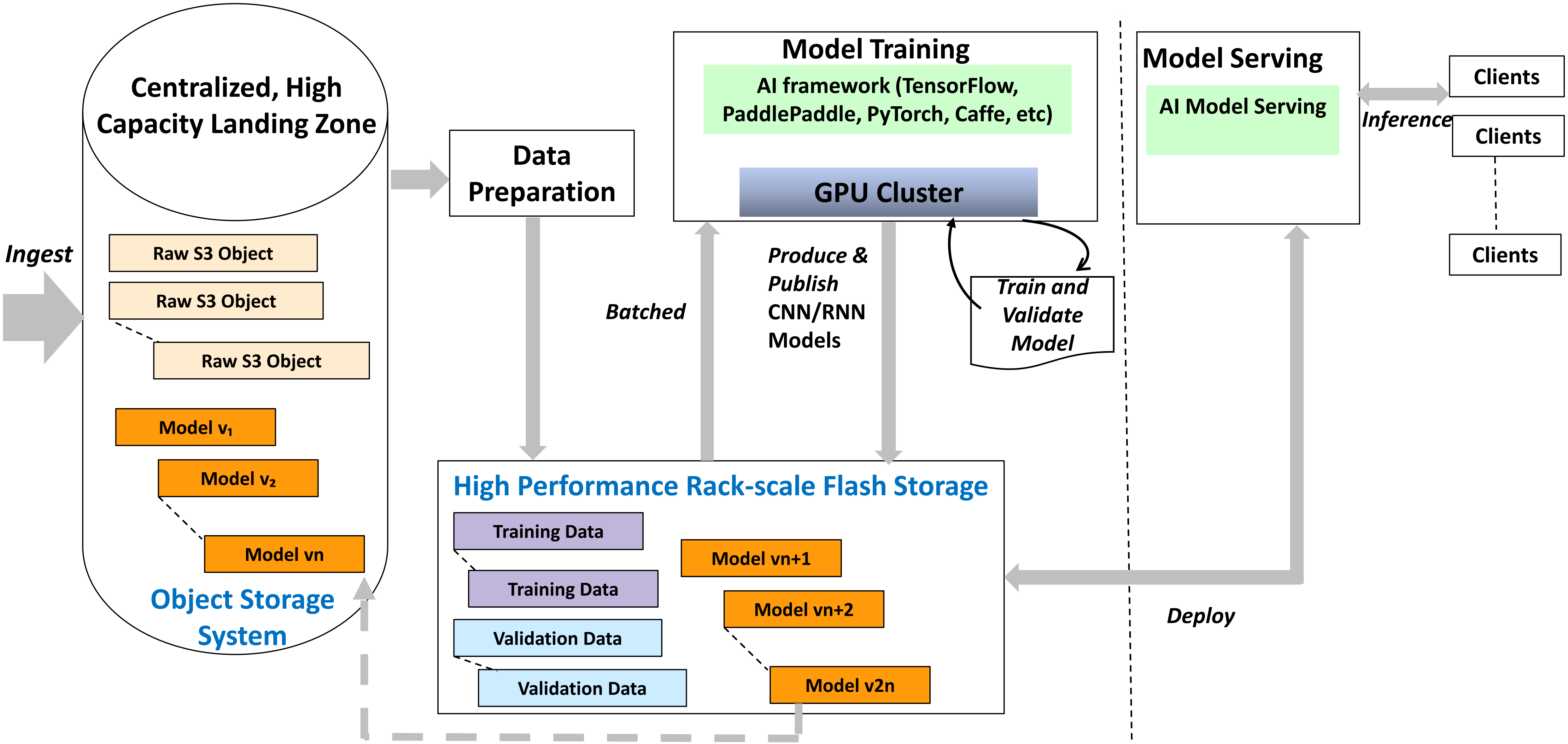


- **Model training is limited to the flash storage capacity integrated in the GPU servers and this incurs multiple data transfers from the object storage, once data grows over capacity on the servers.**
- **Incurs delays in model serving and inference.**

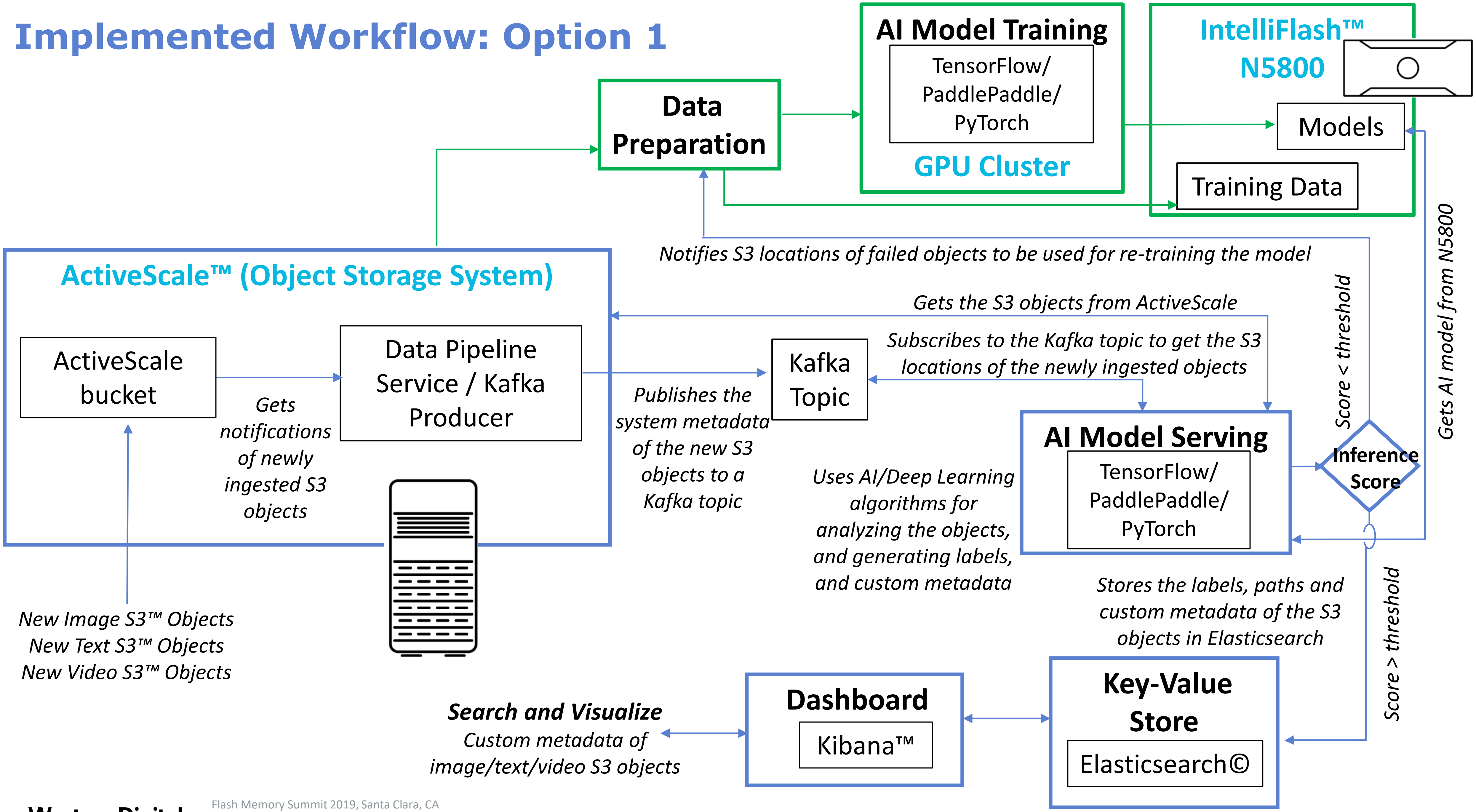


- **Model training can scale independently on a disaggregated pool of GPUs, shared flash and object storage, with no subsequent data transfers.**
- **Inference by the model serving client is faster due to immediate access to trained models on the shared flash storage.**

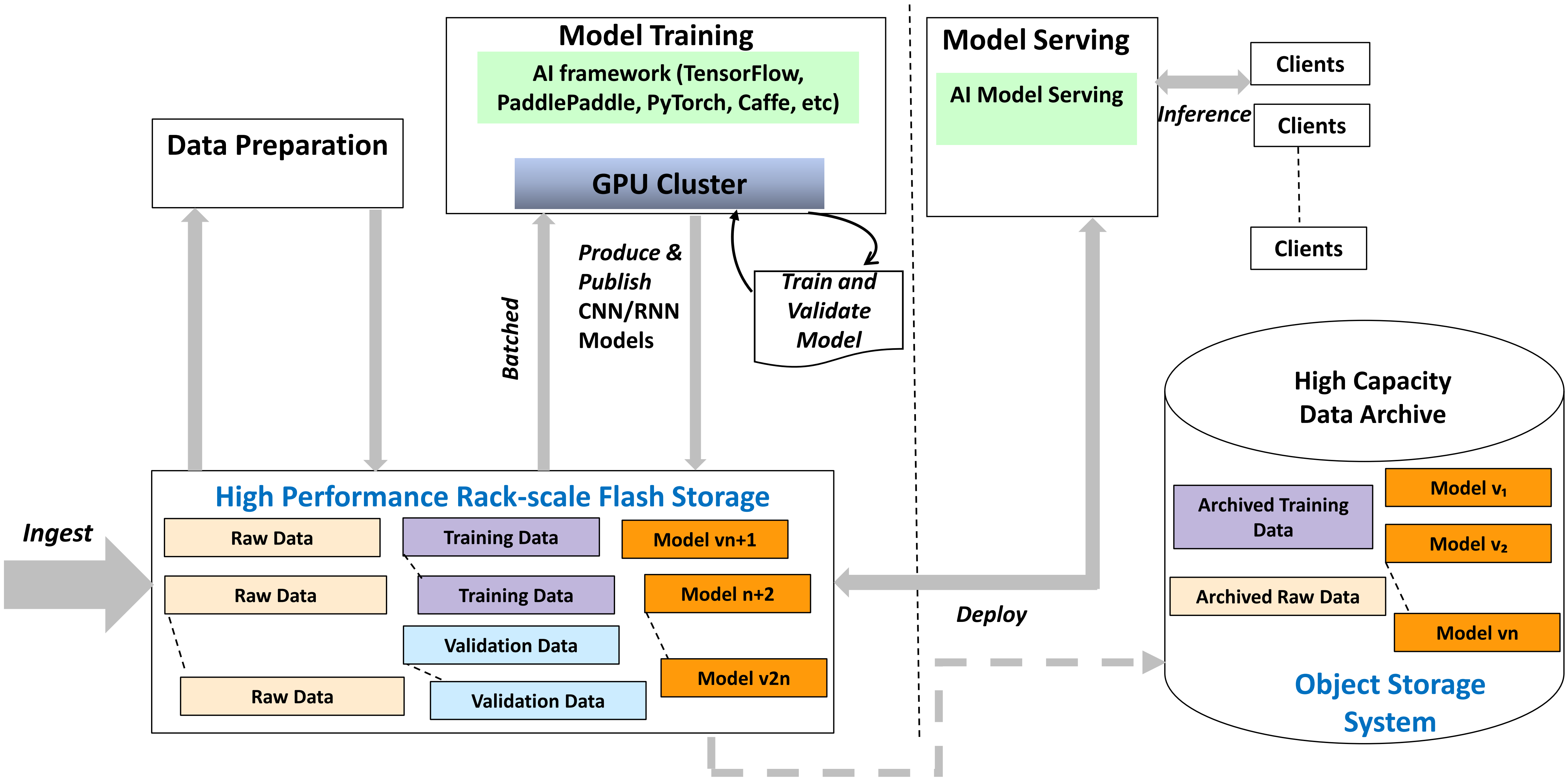
Option 1: Disaggregated Architecture of an AI Data Pipeline



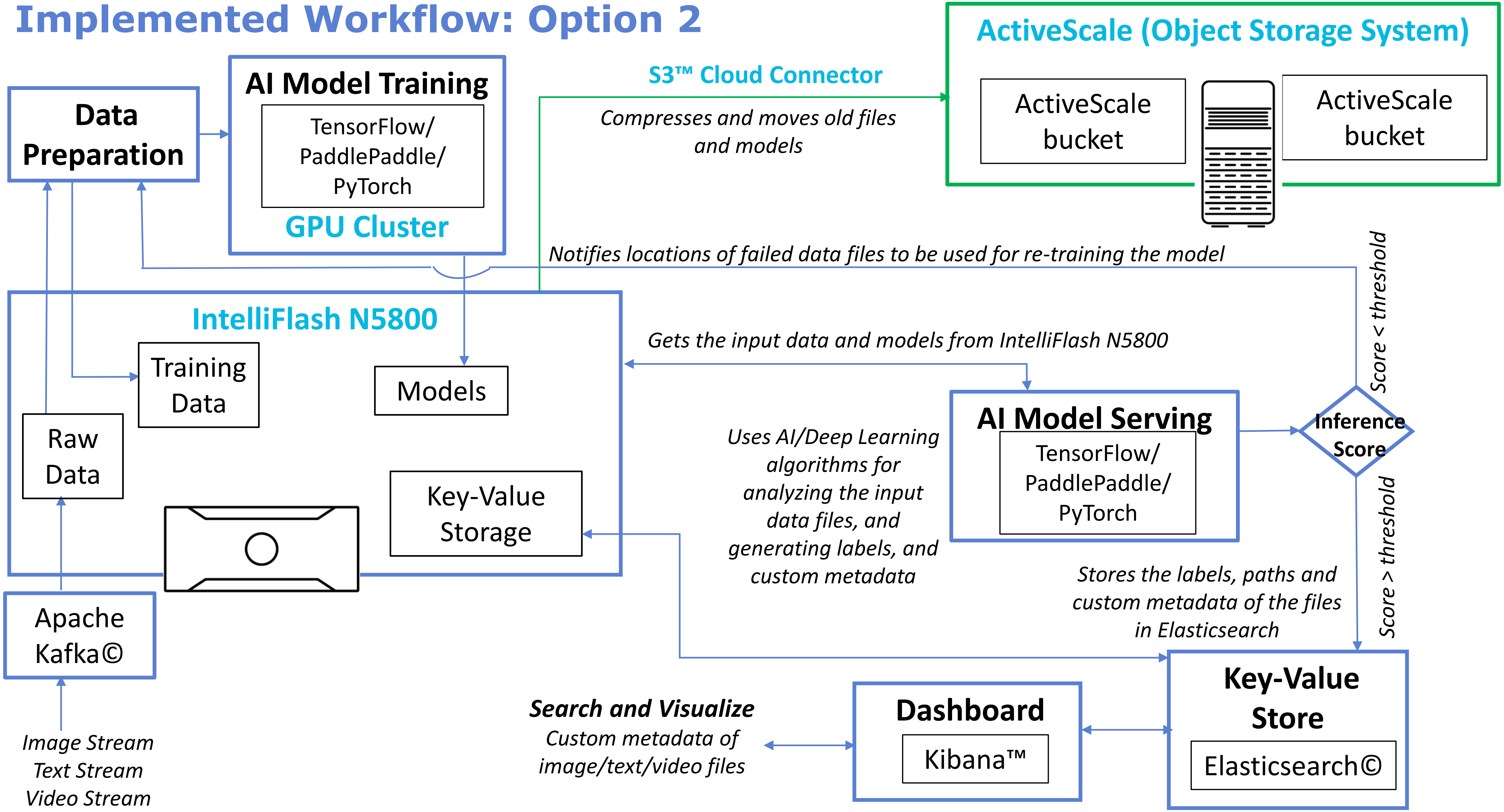
Implemented Workflow: Option 1



Option 2: Disaggregated Architecture of an AI Data Pipeline



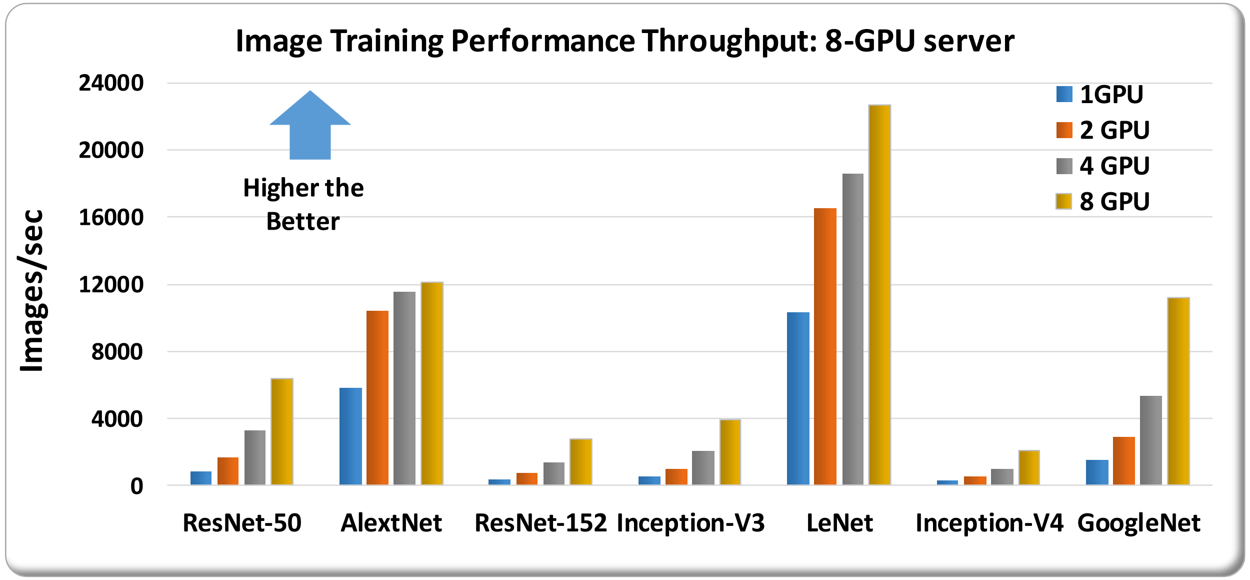
Implemented Workflow: Option 2

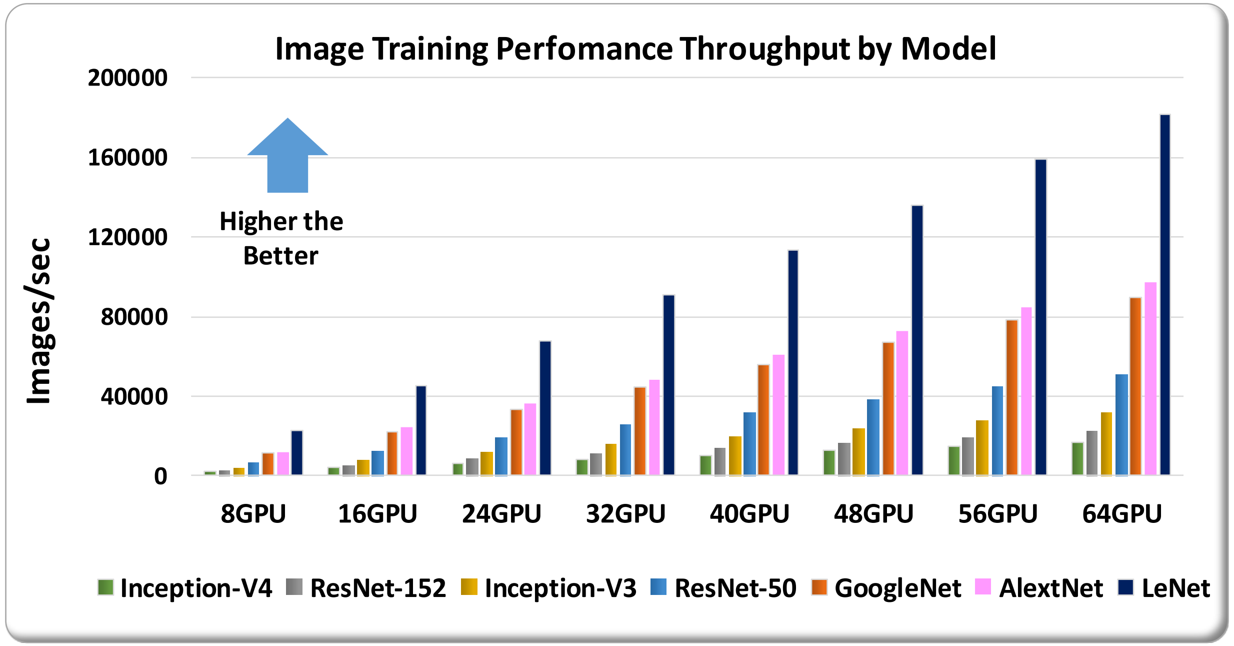


Topics Covered

1. An AI Data Pipeline - Stages and Requirements
2. Disaggregated Architectures for AI Implementations
3. Training and Inference Performance on a Disaggregated Architecture
4. Data Ingestion Performance with Apache Kafka© to an NVMe™ All-Flash Array
5. A Real-World Use Case
6. Summary and Best Practices

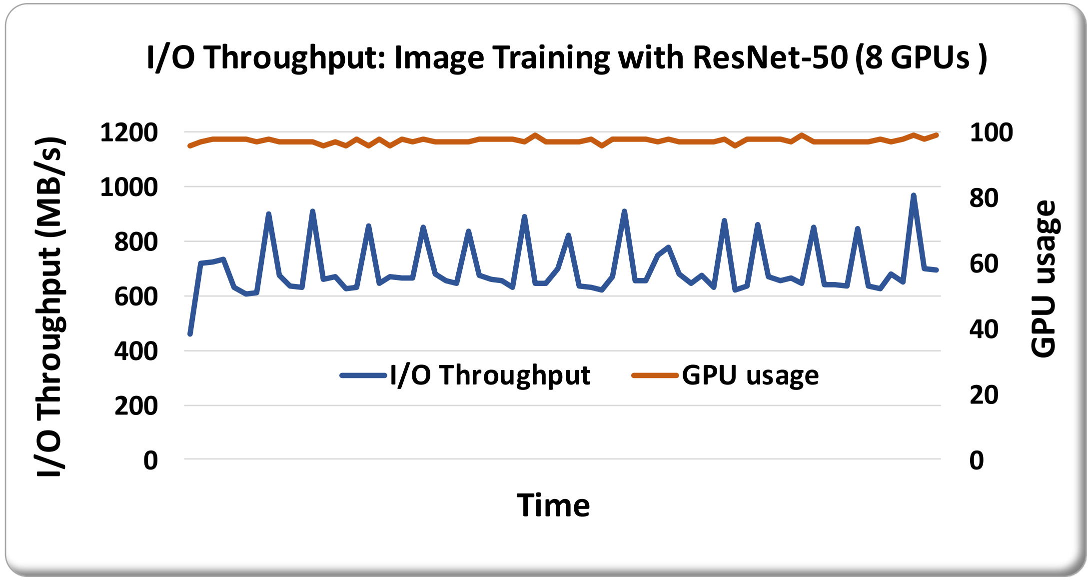
Image Training Performance: Disaggregated Flash and GPUs

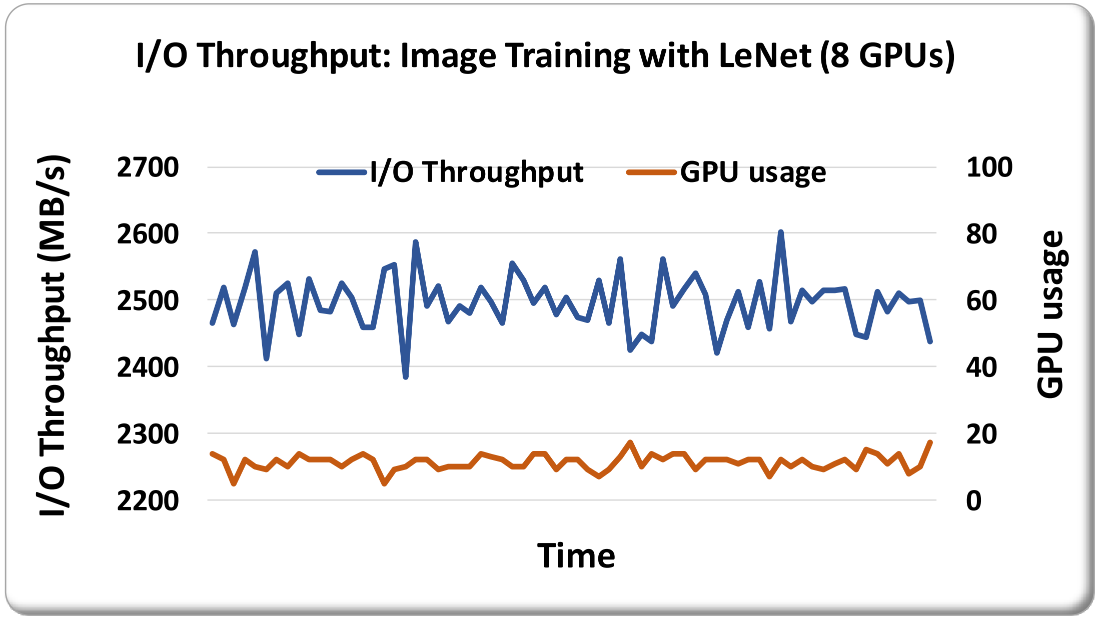




- On a disaggregated architecture comprising an NVMe all-flash array, and
  - a single 8-GPU server, the training performance with most AI models scales almost linearly up to 8 GPUs, except for AlexNet and LeNet, where training performance scales linearly up to 2 GPUs.
  - multiple GPU servers, the training performance scales linearly with the number of servers, irrespective of the choice of AI models.

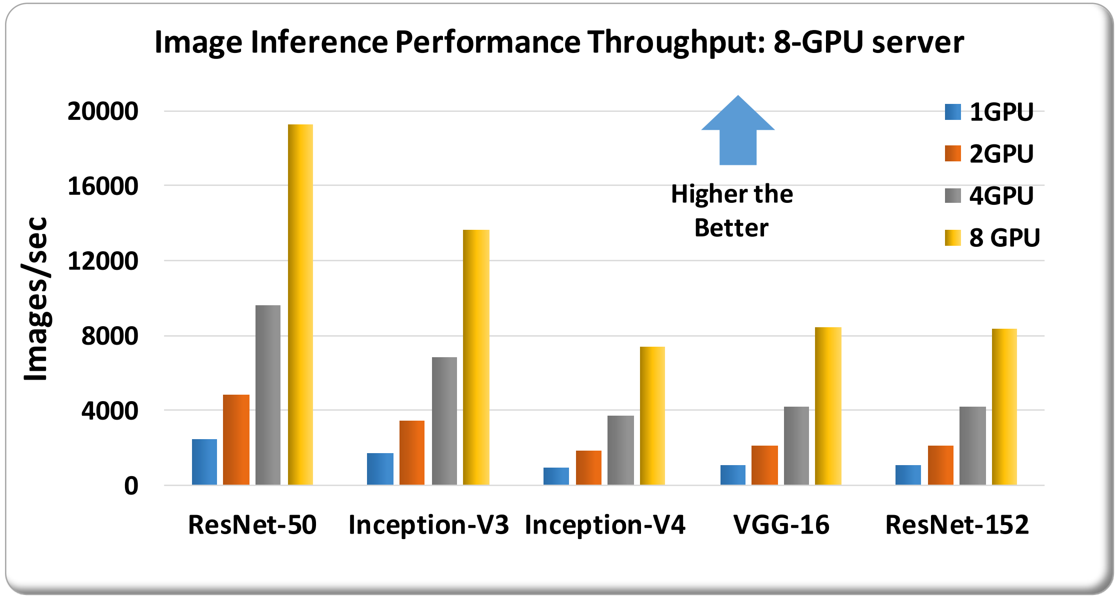
I/O Throughput: Image Training on Disaggregated Flash and GPUs

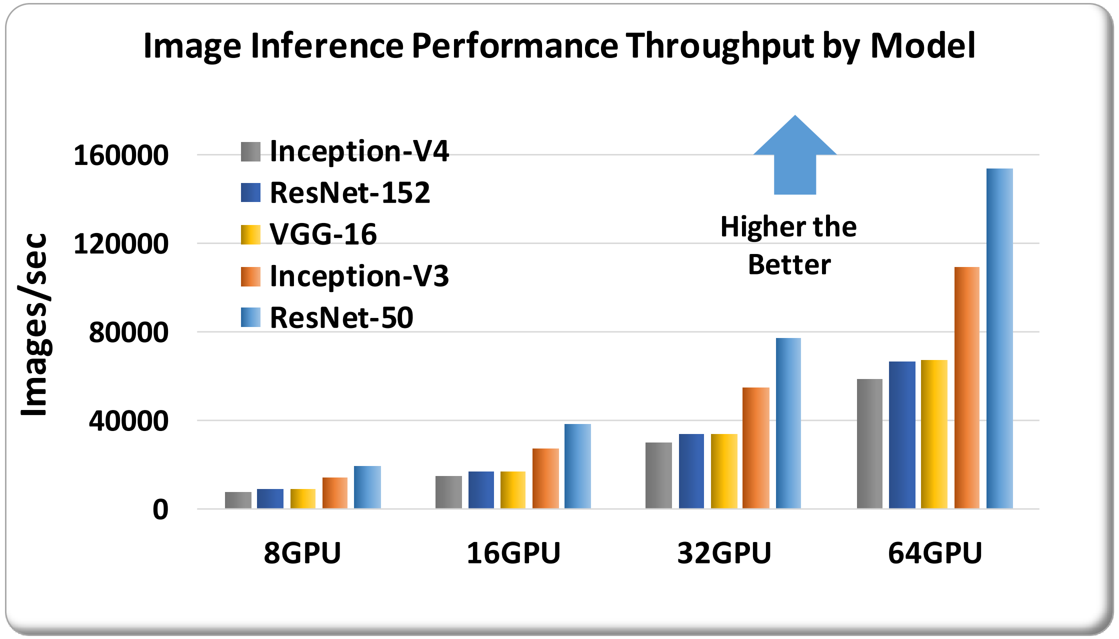




- On a disaggregated architecture comprising a single 8-GPU server and an NVMe all-flash array –
  - the average I/O throughput during training using the **ResNet-50 model (compute intensive)** is ~800 MB/s, the GPU utilization being 97-100% (size of image data is 164 GB, each image being ~100 KB)
  - the average I/O throughput during training using the **LeNet model (I/O intensive)** is ~2.5 GB/s, the GPU utilization being 17-20%. So the **LeNet model yields ~3x the I/O throughput, compared to ResNet-50.**

Image Inference Performance on Disaggregated Flash and GPUs





- The inference throughput is measured as the aggregated images/sec inference results using ImageNet datasets across multiple GPU containers.

- On a disaggregated architecture comprising an NVMe all-flash array, and
  - a single 8-GPU server, **results show that the inference image processing rates are between ~3x to ~3.5x the training rates of the corresponding TensorFlow models.**
  - multiple GPU servers, **users have the flexibility to run mixed AI workloads for training and inference, by dedicating one or two GPUs to inference for every 8 GPUs, rest being allocated to training.**

Example Configurations: GPU servers and an IntelliFlash N5800 Array

- An example allocation strategy of IntelliFlash N-series arrays is considered for executing AI workloads –
  - **30% of the I/O bandwidth for model training, and the remaining 70% for various phases like data preparation, inference, and other activities**
- **Considering the above allocation strategy**, example configurations are derived using the I/O throughput achieved on a disaggregated architecture comprising a single IntelliFlash N5800 array and a 8-GPU server, while using ResNet-50 and LeNet models for training -
  - **A single IntelliFlash N5800 array can scale up to nine 8-GPU servers running ResNet-50 model for the training phase, with 100% utilization of GPUs.**
  - **With LeNet model, a single IntelliFlash N5800 array can optimally scale up to three 8-GPU servers for the training phase.**

	IntelliFlash N5800
Number of Arrays	1
Mixed I/O Throughput GB/s (80% reads, 20% writes)	23.5
Number of 8-GPU servers (using compute-intensive ResNet-50 model for training)	9
Number of 8-GPU servers (using I/O-intensive LeNet model for training)	3

Topics Covered

1 An AI Data Pipeline - Stages and Requirements

2 Disaggregated Architectures for AI Implementations

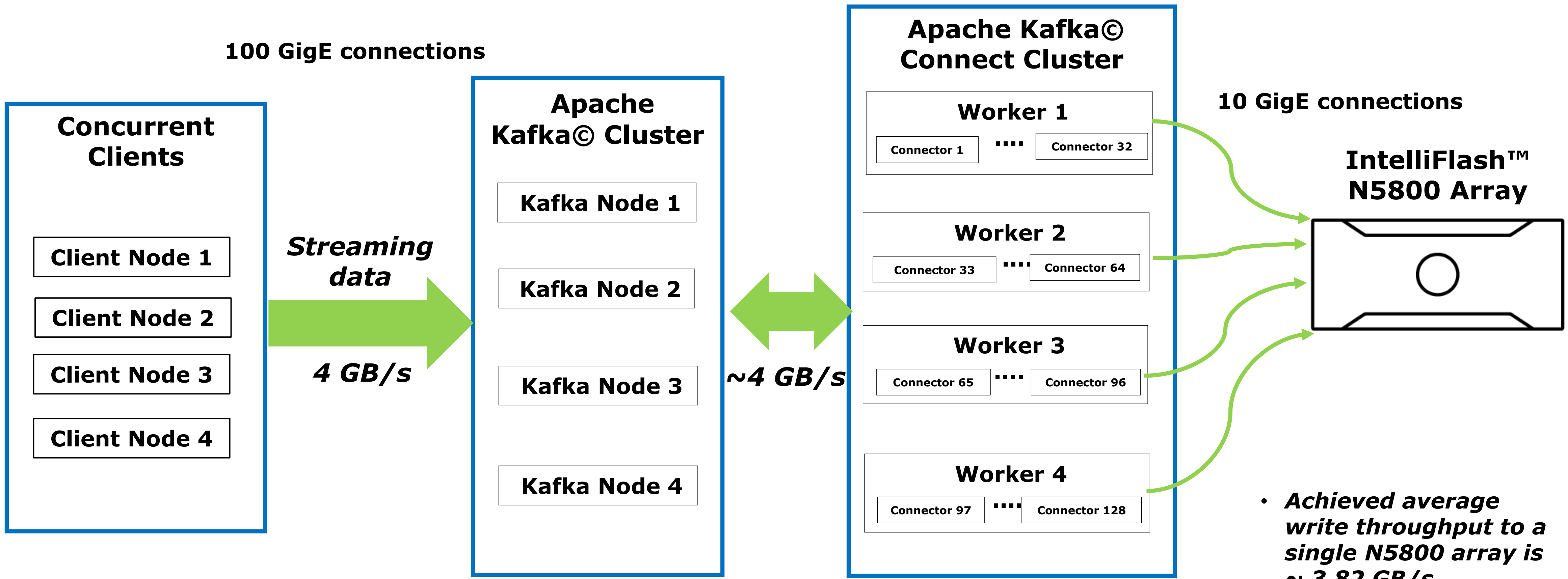
3 Training and Inference Performance on a Disaggregated Architecture

4 Data Ingestion Performance with Apache Kafka© to an NVMe™ All-Flash Array

5 A Real-World Use Case

6 Summary and Best Practices

Data Ingestion Benchmark Schematics



For each benchmark run on 4 clients:

- ***Record size: 500 bytes/message***
- ***Sends 400 M messages at 4 GB/s***
- ***Publishes to a Kafka topic, 8 M messages at a time.***

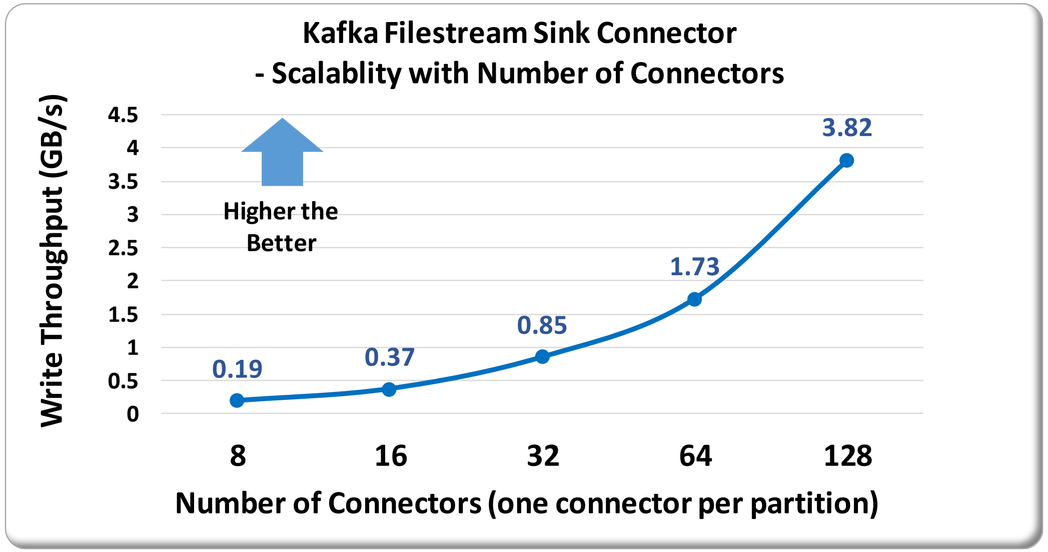
For each benchmark run, Kafka cluster can process

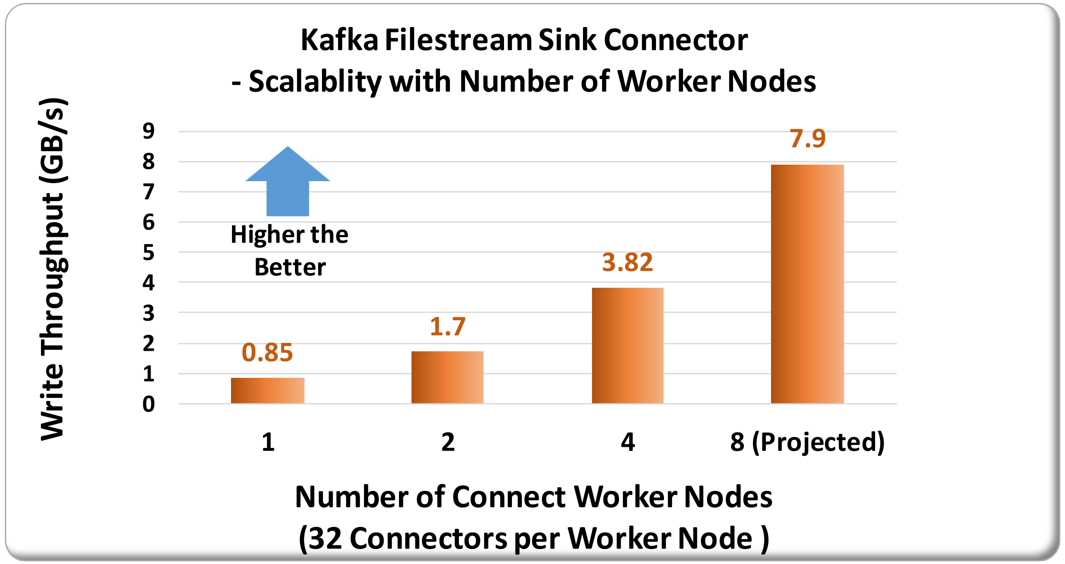
- ***8 M messages/sec, for the message size of 500 bytes.***
- ***Average latency of acknowledgment to the clients is ~40 ms/message.***

- ***Each connector in the Kafka Connect Cluster subscribes to a Kafka topic in the Kafka Cluster, and sinks the data to the N5800 array.***
- ***A total of 128 connectors is writing to a single N5800 array.***

- ***Achieved average write throughput to a single N5800 array is ~ 3.82 GB/s.***
- ***Projected throughput will increase to 7.9 GB/s for higher ingestion rates, while using 8 worker nodes in the Kafka Connect cluster.***

Data Ingestion Performance with Kafka to an NVMe All-Flash Array





- Each connector (known as sink connector) is assigned to a partition of the respective Kafka topic, i.e., the number of connectors is equal to the number of partitions/files.
- **Write throughput increases linearly with the number of connectors.**
- With a single IntelliFlash N5800 array, 128 sink connectors and 128 Kafka partitions, a 4-node Kafka Connect cluster provides a **write throughput of 3.82 GB/s, for an ingestion rate of 4 GB/s.**
- **A maximum of 7.9 GB/s write throughput with a single IntelliFlash N5800 array can be achieved with 8 (projected) Kafka Connect worker nodes, for ingestion rates higher than 4 GB/s.**
- **This test helps to determine the number of connectors to configure in the Kafka Connect cluster, based on the number of N5800 arrays, the input ingestion rates, and the available I/O throughput from the flash arrays.**
- The CPU usage is 80% per worker node (having 32 connectors) in the 4-node Kafka Connect cluster to achieve a max throughput with 128 connectors.
- 4 JVMs are used for each worker node in the Kafka Connect cluster, with a JVM heap size of 64 GB.

Topics Covered

1 An AI Data Pipeline - Stages and Requirements

2 Disaggregated Architectures for AI Implementations

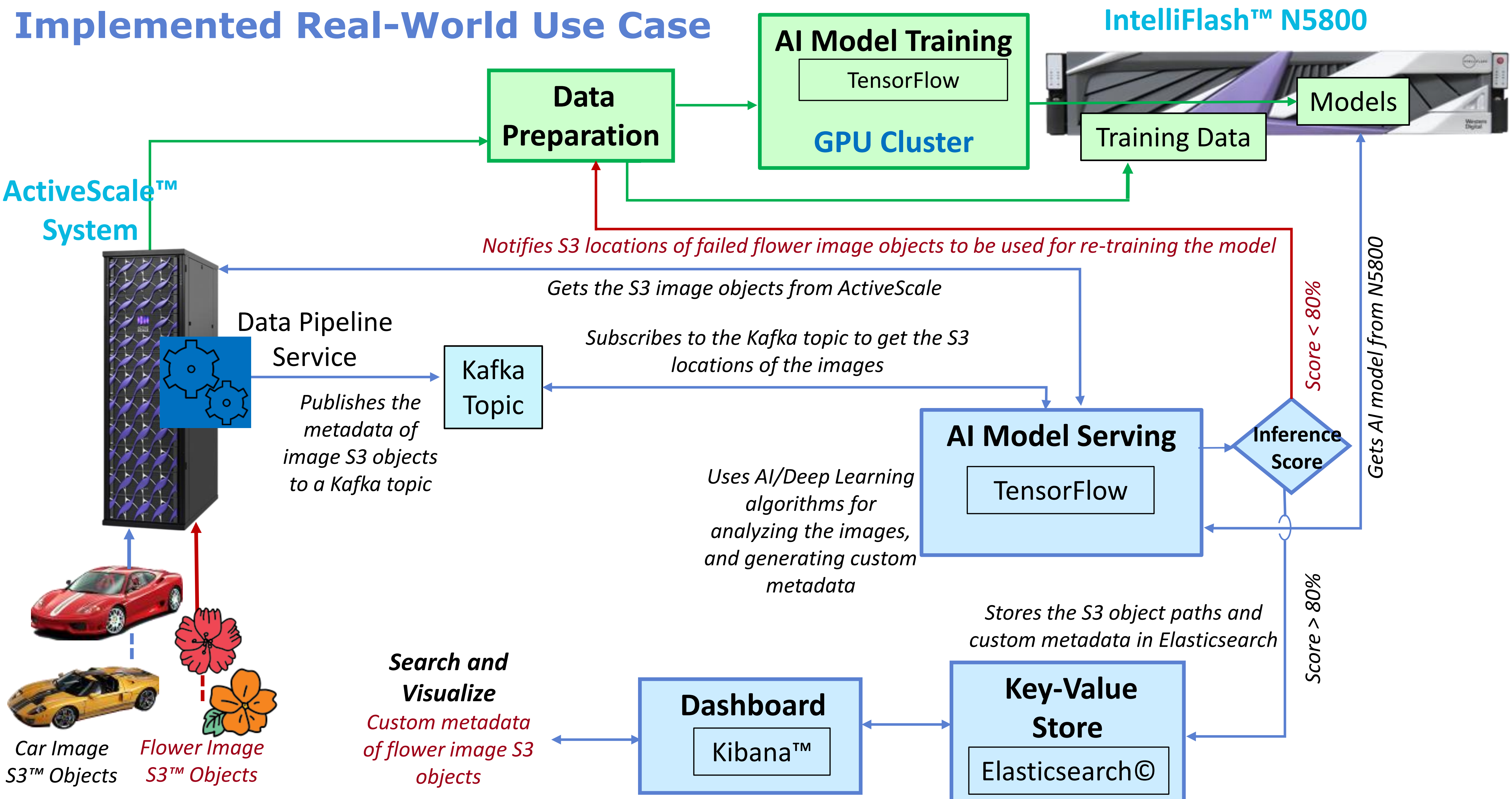
3 Training and Inference Performance on a Disaggregated Architecture

4 Data Ingestion Performance with Apache Kafka© to an NVMe™ All-Flash Array

5 A Real-World Use Case

6 Summary and Best Practices

Implemented Real-World Use Case



Topics Covered

1. An AI Data Pipeline - Stages and Requirements
2. Disaggregated Architectures for AI Implementations
3. Training and Inference Performance on a Disaggregated Architecture
4. Data Ingestion Performance with Apache Kafka© to an NVMe™ All-Flash Array
5. A Real-World Use Case
6. Summary and Best Practices

Summary and Best Practices

- **Implementing a disaggregated architecture of GPU compute, a shared pool of IntelliFlash N-series arrays and ActiveScale system(s) has multiple benefits while executing AI workloads –**
  - **Subsequent data transfers in and out of local SSDs of GPU servers can be avoided, as the data grows over capacity.**
  - **Inference is faster due to immediate access to trained models on the shared flash storage.**
  - **Businesses have the ability to scale GPU servers and shared flash arrays independently to meet the changing needs of their AI workloads.**
  - **Users have the flexibility to run mixed AI workloads for training and inference.**
  - **With a preferred allocation strategy of the I/O bandwidth, various teams can share and scale the IntelliFlash N-series arrays to serve multiple GPU servers in a cost-effective manner.**
  - **A high capacity object storage system like ActiveScale, as a component of the disaggregated architecture, may be used as a landing zone for the ingested data as well as an archival solution.**
- **As a best practice to attain an optimal ingestion performance with Kafka to IntelliFlash N-series arrays, tuning the following parameters is recommended -**
  - **Number of connectors and worker nodes in the Kafka Connect cluster, based on the number of N-series arrays, the input ingestion rates, and the available I/O throughput from the arrays;**
  - **Based on the I/O throughput requirement, high-speed network interfaces and topology need to be configured for the Kafka cluster, the worker nodes of the Kafka Connect cluster, and the IntelliFlash N-series array(s) to eliminate network bottlenecks.**

Western Digital®

Architecting Data Infrastructure for the Zettabyte Age