STORAGE-OPTIMIZED MACHINE LEARNING

IMPACT OF STORAGE ON MACHINE LEARNING

SUMMARY

Organizations run at the speed of their data. While Artificial Intelligence (AI) and Machine Learning (ML) have a continuing history of solving traditional problems in pattern recognition, AI and ML techniques are rapidly finding their place in business analytics, where the patterns being determined might be less obvious. The efficiency of these learning systems can define an organization's competitive advantage.

Machine Learning has long been implemented on top of traditional compute architectures, where throughput and latencies are determined by coupling compute and storage through the same networking and storage interconnects that serve other business applications. The increasing volume and velocity of arriving data are stressing these architectures, whether for real-time processing of Internet-of-Things (IoT) telemetry, pattern recognition in images or audio, or mining data from the warehouse to gain new insights about an organization's customers or business.

This paper describes how current implementations for machine learning and related processing can be built atop new storage architectures specifically designed to deliver data in real-time to provide a storage-optimized solution to the deep learning problem.

THE BUSINESS OF MACHINE LEARNING

DATA IS DISRUPTIVE.

The volume and velocity of enterprise data is rapidly expanding. Data is flowing in from new, non-traditional, sources, which increasingly live outside of the datacenter. Data sources as disparate as social media and IoT are flooding the enterprise with an overwhelming number of bits. These bits can lay dormant on archive drives, or can be turned into information that can be leveraged to beat the competition.

At the same time, organizations are finding new uses for the contents of their existing data warehouse. Mining the many petabytes of stored data, much of it long thought dormant, is yielding new insights and business opportunities.

This is increasingly being accomplished through the use of machine learning algorithms, where the techniques long-practiced in the artificial intelligence world are being leveraged to find new patterns. These patterns lead to insights which increasingly touch every aspect of an organization's business.

Machine learning does more than provide image or voice recognition. It optimizes the supply chain. It allows an enterprise to mine customer transaction data to gain insights about new markets and opportunities. Machine learning identifies patterns in health care and the sciences. It powers the conversations between humans and automated response systems. Machine learning puts intelligence in the devices we carry and the cars that we drive. It's everywhere and is becoming more pervasive.

MACHINE LEARNING IN TODAY'S WORLD

WHAT IS MACHINE LEARNING?

Stated simply, Machine Learning is the application of special algorithms that are adept at identifying patterns in datasets. The more data there is for these algorithms to explore, the better the results tend to be. Data from which patterns should be extracted are run through these algorithms in a process called *training.* Once patterns emerge, the results can be used to identify similar patterns in new datasets. Voice recognition algorithms, for example, can be trained on massive audio datasets to determine the right patterns to recognize useful words.

While variations exist, there are two basic approaches to the training process.

First, there is *supervised learning*, where human intervention is required to direct the desired outcome of learning. Imagine the act of training a computer to recognize the picture of a cat. The algorithms can be fed tens of thousands of variations of cat images, along with some images that aren't cats, in order to instruct it to recognize the feline animal when presented with a picture of it in the future. Supervised learning relies on human (or machine) labeled datasets to tell the system when it's a cat image, and when it isn't, so that it can learn.

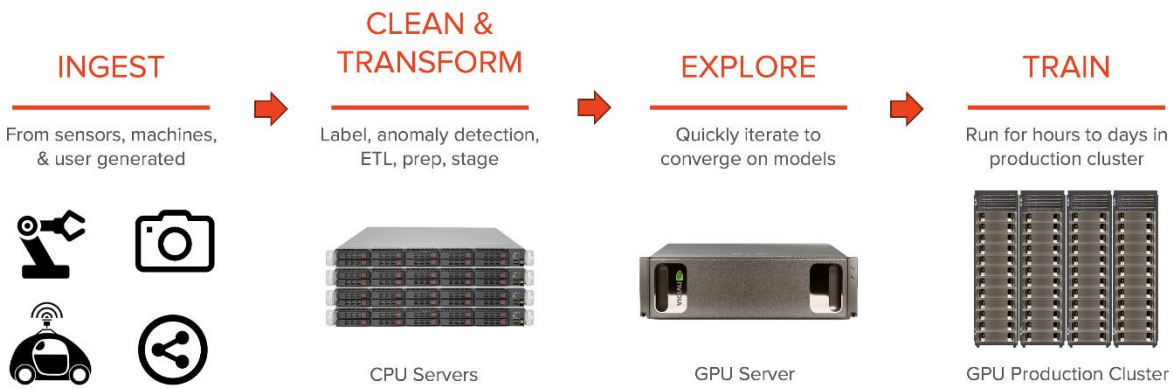
Second, is *unsupervised learning*. As its name suggests, unsupervised learning explores unlabeled datasets and identifies patterns that may not be obvious to humans. This is an area that is extremely active in the business world today, as unsupervised

learning can find patterns in customer transaction data, health data, financial trading and many others.

Despite the differences in how learning might work for a given dataset and desired outcome, there are very common traits in how systems are built to ingest data, learn from it, and store it to learn again in the future.

The following figure nicely demonstrates the complexities involved in the process from a system workflow perspective.

FIGURE 1: MODERN LEARNING PIPELINE



Source: Pure Storage

Machine Learning Workflows.

The first step is *ingesting* the data that the training algorithms will examine. Data is brought into the workflow from outside. This could be from any number of sources. Maybe it's real-time data, from an array of sensors, or it could be data generated by business operations and stored in the traditional storage arrays in the enterprise datacenter. Often, it is data that was used in learning previously, and tested again by the training algorithms as they've evolved. In all of these cases, the data is moved from its source to persistent storage somewhat closer to the learning environment.

Before being analyzed by the learning algorithms, the ingested data needs to be prepared. This is the *cleaning* stage. Algorithms may be run on it to detect anomalies, remove duplicate data, enrich the data with meta-data tags, or perform any number of other tasks based on the goals of the analysis. This is data intensive, but not always computationally taxing. The speed of the process depends on the efficiency of moving the data through the cleaning process.

Once data is ingested and cleaned, the training algorithms are almost ready to execute. Often times a number of experiments will be run on the datasets to help set the stage for the actual learning process. This is the *exploration* phase, where researchers work with the data to understand and help guide the learning process to come.

Finally, there is the *training* stage. There are numerous variations on what happens during this phase. The process can take anywhere from hours to weeks depending on the goals of the learning. The important thing to note is that it is computationally taxing, and the efficiency (and duration) of the process is a combination of the effectiveness of the compute engines, coupled with the ability to keep relevant data fed into them.

THE SHAPE OF DATA

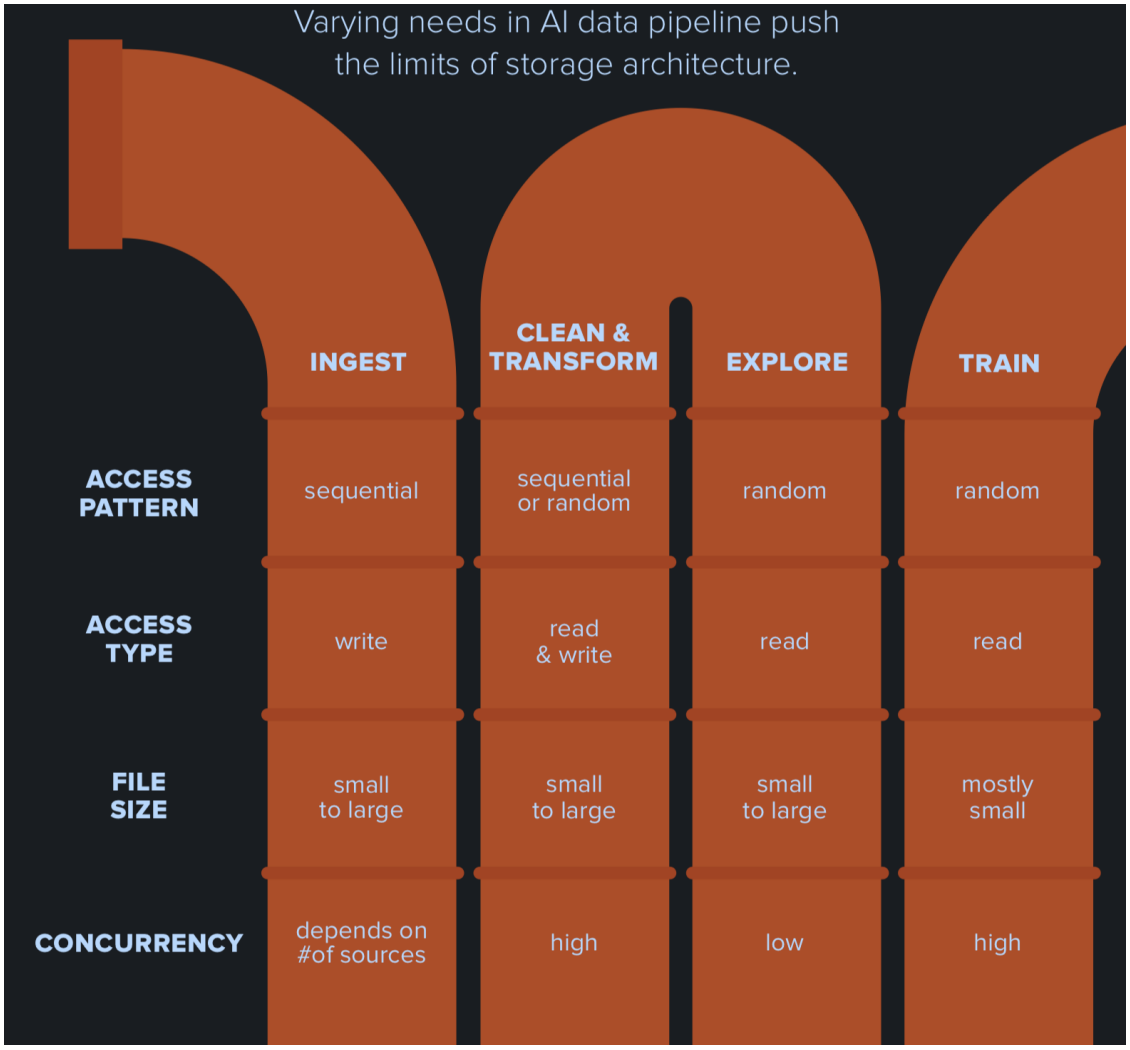
While thinking about the four basic steps of machine learning, it's instructive to think about the shape and characteristics of the data as it traverses the workflow. This is illustrated in figure 2.

Ingestion is fairly simple. Data enters the workflow. This data tends to arrive as sequential writes of various sizes (depending on the source). This means the data streams into a storage array at a relatively unchallenging rate.

Cleaning becomes more taxing on the data. The access patterns become more unpredictable, generating both reads and writes to the disk, in both random and sequential patterns of varying sizes. Cleaning and preparation tends to be as taxing on the storage systems as it is computationally. Cleaning touches all of the data, and in unpredictable patterns. The underlying storage must be matched to activity.

Exploration and training are both very read-heavy processes. As the data is explored and churned on by the training algorithms, it's read over and over again. The access patterns are not predictable in a reasonable sense, and tend to be very small transactions from the storage systems feeding them. Training requires many small objects. The efficiency of the algorithms is directly tied to the effectiveness of the persistent storage underlying the process.

FIGURE 2: TYPICAL MACHINE LEARNING PIPELINE



Source: Pure Storage

CHALLENGES OF MACHINE LEARNING AND STORAGE

EVOLUTION OF COMPUTE.

It requires a tremendous amount of computational horsepower and data volume to process today's machine learning. It wasn't very long ago that traditional x86-based servers, backed by traditional storage arrays and network attached storage (NAS) devices, were good enough to run learning algorithms. These are, after all, the same servers and storage networks that enterprises trust to run their other business critical applications.

The recent revolution in machine learning, driven by deep learning, occurred due to a number of converging innovations. Technology arrived at a point where it could begin to deliver on the promise of artificial intelligence and machine learning in a material way. The industry changed the art of the possible.

DRAM became relatively cheap and plentiful. In-memory computing models rapidly evolved to where, instead of fetching data from external storage, streaming data is fed into the same domain as the processors performing the analytics. This allowed solutions such as Apache Kafka, which consolidates a number of data sources to proxy to the analytics clusters, to evolve.

At the same time, the graphics processing unit (GPU) became about more than just delivering fast action to first person gamers. It turns out that the parallel processing units in these GPUs are *really* good at executing exactly the kinds of algorithms required to satisfy the demands put forth by machine learning.

While fast processors are great, GPUs have historically been difficult to program. As a leader in GPU innovation, NVIDIA worked to lessen the burden of programming GPUs by innovating a language called CUDA and a wide range of tools to help the data scientists who develop the training algorithms. The GPU software ecosystem has blossomed, becoming the enabler for a constellation of new deep learning algorithms.

To provide an idea of the current state-of-the-art in GPU compute, see Figure 3 with data from NVIDIA, an innovator of high-performance GPUs. The graphic demonstrates that NVIDIA's top-tier GPU-based solutions can train on certain algorithms up to 96 times faster than a traditional CPU-only solution. A much more modest solution, with eight mid-tier GPUs, can train 18 times faster. It's safe to say that almost all modern machine learning implementations rely heavily on GPU-based compute.

FIGURE 3: GPU PROCESSING VS TRADITIONAL PROCESSING

Source: Moor Insights & Strategy

WAITING FOR I/O.

Efficient machine learning is not all about the capabilities of the compute engine. Modern compute, whether CPU or GPU, is fast, demanding more data at a faster velocity than traditional infrastructure's capability to deliver.

The existing data may be training data for image or other pattern recognition, or it could be customer transaction or market data kept in an enterprise data warehouse. Either way, it has similar properties when thinking about the implementation of a learning system.

Machine learning based on real-time and other non-batch training data is often gated by the front-end work of cleaning and preparing the data that will be fed to the learning algorithms. The prepared data is put onto persistent storage, where it is ultimately consumed by the compute engines responsible for the learning algorithms. This is true to a lesser extent for warehoused, or other currently existing data. Existing data may have already been prepared.

The latencies and throughput of the storage systems housing the data directly impact the performance of the system. While it's commonly assumed the interconnect between systems may be the bottleneck, Facebook recently published research showing that today's Ethernet-based networks are sufficient in providing near-linear scaling capabilities (link: https://research.fb.com/wp-content/uploads/2017/12/hpca-2018-facebook.pdf). Data bottlenecks lie in the storage layer, and the storage system's capability to deliver tremendous throughput at low latencies.

Compute nodes can often suffer from data inefficiencies and starvation due to high latencies from storage arrays. Latency is the amount of time that it takes for a transaction between two devices to complete. That transaction might be communication between two compute nodes across a network link, or between a compute node and a storage device, which is feeding that compute node new data to process.

During a fetch of data from a remote storage device, for example, the overall latency is the product of a number of operations. Data must traverse the interconnect between the processing node and the storage device, resulting in a certain amount of latency.

The speed of the disk, which ultimately serves up the data, has significant impact on the amount of time it takes to respond to the transfer requests. Traditional platter-based hard drives require a request to wait for a number of mechanical operations to occur,

while SSD-based storage systems typically respond very close to the speed of the silicon supporting the storage, lowering the latency. However, many storage vendors using off-the-shelf SSDs quickly hit performance limitations due to serial interfaces to the SSDs and legacy software bolted on top of the system to manage the SSDs.

Complicating things even further are the access patterns inherent in machine learning, discussed previously. While ingestion is a relatively straight-forward streaming of sequential writes, the remainder of the process tends towards very small random reads.

Intelligent storage systems compensate for the relative slowness of the underlying persistent storage media (such as spinning hard disks, or even SSDs) by attempting to discern patterns in data access and fetching that data before it's needed. The more random the pattern, the harder that is to do.

ARCHITECTING FOR EFFICIENCY

OPTIMIZING STORAGE.

There is a lot of involved in something as seemingly simple as moving a block of data from a disk drive to a compute cluster, all of which can conspire to slow down the processing. This is not an intractable problem. It can be avoided.

As we've seen, the size and locality of the reads and writes to a storage system have a major impact on the ability of that storage systems to behave efficiently for the application that it's servicing. A storage system can be tuned to meet the needs of machine learning. It's important to note that those needs may be very distinct from most other business applications

There are three major factors that influence the ability of a storage system to provide fast and effective data to a machine learning infrastructure:

1. **Locality of data**. One of the biggest causes of latency is the amount of time that it takes to bring data from a storage device to the processor that will consume it. Locating data near the machine learning cluster that will consume it is a necessity, which is one of the major problems of leveraging public cloud for machine learning. Utilizing an array that is capable of spreading its data across a large number of storage processing devices will drive latency down even further, while also having a net effect of increasing overall throughput.

2. **Access Patterns**. This paper has discussed the challenges faced by a storage system in predicting and optimizing for traffic patterns unique to machine learning. Small block, unstructured data being accessed randomly is the norm for machine learning. This type of access pattern has historically been the most difficult design point for any storage system to meet. A storage array that can optimize itself to respond to those patterns is a key requirement of any machine learning architecture.

3. **Capacity**. Machine learning thrives on data. The more data, the better the results. Moving data between multiple storage systems and the compute elements hosting the machine learning algorithms is a major impactor on overall efficiency. At the same time, machine learning tends to breed new appetites for data. Future proofing with elastic storage capabilities delivered through a single footprint is a demonstrable benefit for machine learning implementations.

Additionally, it's increasingly common for machine learning solutions to consume data using object storage semantics. Object storage differs from traditional block storage in that data exists within a navigable namespace, which allows applications to quickly identify and consume those objects. Object storage differs from a traditional filesystem in that it both exists on the storage device and is a very flat pool of data. Today, most object store systems are optimized for long-term, archival store use cases, not for high performance.

CHOOSING SUCCESS.

We've discussed a number of factors that influence how storage impacts machine learning. Is there a solution? Yes, there is.

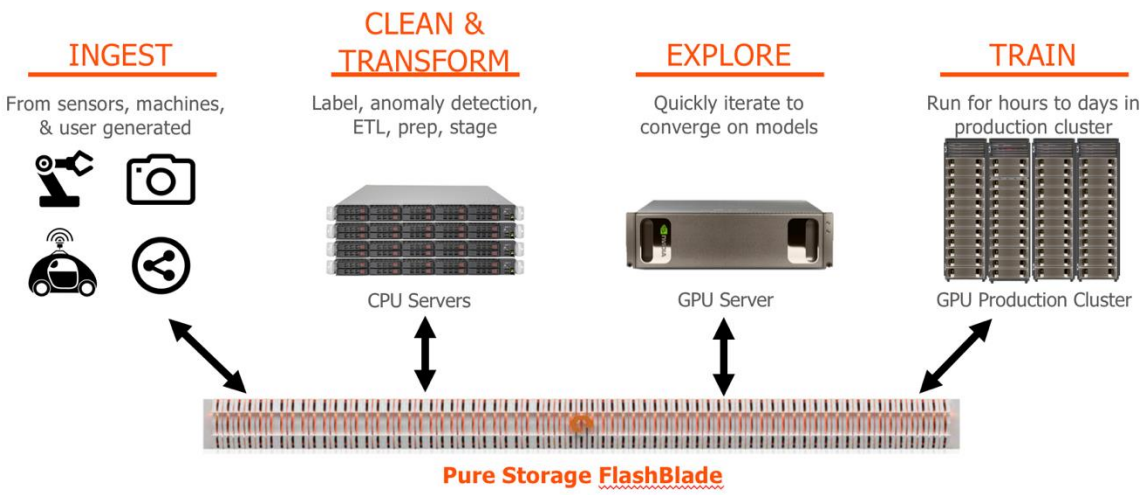
Figure 4 illustrates a storage hierarchy for machine learning to address these factors. At the center of the architecture is the Pure Storage FlashBlade. Flashblade provides primary storage for the entire machine learning workflow, laying a foundation for data-centric infrastructure for many enterprise customers deploying AI in production.

FlashBlade is optimized for the storage needs of machine learning, providing high performance file and object storage, scalable elastic storage and massive throughput for any access pattern, sequential or random, up to (as Pure Storage reports) 75GB/s from a cluster of devices.

Similar to what parallel GPU processors did for the computing industry, FlashBlade is the first in the storage industry to be architecturally massively parallel from the ground

up. While storage systems generally fail at small (50K) random file I/O, FlashBlade is optimized to process and deliver small, meta-data heavy workloads as well as large, sequential I/O. It also supports native object store (leveraging Amazon's S3 semantics, which has become the de facto standard for object store).

FIGURE 4: NEW STORAGE HIERARCHY



Source: Moor Insights & Strategy

CONCLUSION

Machine learning is complex, time-consuming and data heavy. Architects and practitioners implementing the systems that deliver on the promise of machine learning tend to rightly focus on the complex task of integrating CPUs and GPUs to support the algorithms that will make their endeavors successful.

Storage and delivery of data can dramatically influence the efficiency of a machine learning environment. Innovation in this world continues at a rapid pace, and it's critical that you talk to the leaders who both understand today's machine learning environment and are building tomorrow's.

Solutions such as the FlashBlade from Pure Storage remove the complexity and worry from architecting storage into a machine learning training environment. It provides future-proofing with its elastic file and object storage capabilities, its on-board programmable compute capabilities, and the predictive analytics provided by Pure1.

IMPORTANT INFORMATION ABOUT THIS PAPER

CONTRIBUTOR
Steve McDowell, Senior Analyst at Moor Insights & Strategy

PUBLISHER
Patrick Moorhead, Founder, President, & Principal Analyst at Moor Insights & Strategy

INQUIRIES
Contact us if you would like to discuss this report, and Moor Insights & Strategy will respond promptly.

CITATIONS
This paper can be cited by accredited press and analysts but must be cited in-context, displaying author's name, author's title, and "Moor Insights & Strategy". Non-press and non-analysts must receive prior written permission by Moor Insights & Strategy for any citations.

LICENSING

DISCLOSURES
Pure Storage commissioned this paper. Moor Insights & Strategy provides research, analysis, advising, and consulting to many high-tech companies mentioned in this paper. No employees at the firm hold any equity positions with any companies cited in this document.

DISCLAIMER
The information presented in this document is for informational purposes only and may contain technical inaccuracies, omissions, and typographical errors. Moor Insights & Strategy disclaims all warranties as to the accuracy, completeness, or adequacy of such information and shall have no liability for errors, omissions, or inadequacies in such information. This document consists of the opinions of Moor Insights & Strategy and should not be construed as statements of fact. The opinions expressed herein are subject to change without notice.

Moor Insights & Strategy provides forecasts and forward-looking statements as directional indicators and not as precise predictions of future events. While our forecasts and forward-looking statements represent our current judgment on what the future holds, they are subject to risks and uncertainties that could cause actual results to differ materially. You are cautioned not to place undue reliance on these forecasts and forward-looking statements, which reflect our opinions only as of the date of publication for this document. Please keep in mind that we are not obligating ourselves to revise or publicly release the results of any revision to these forecasts and forward-looking statements in light of new information or future events.