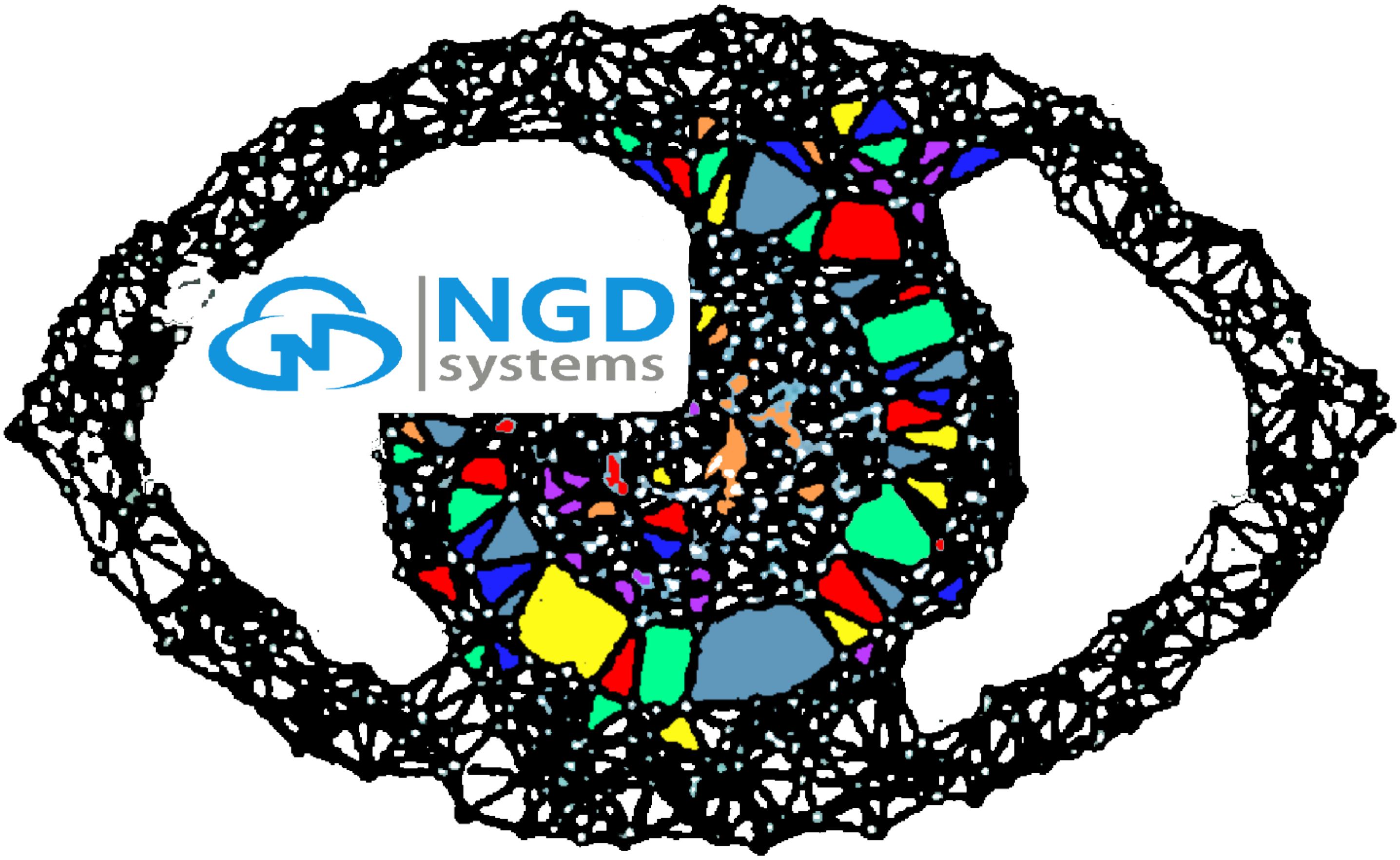
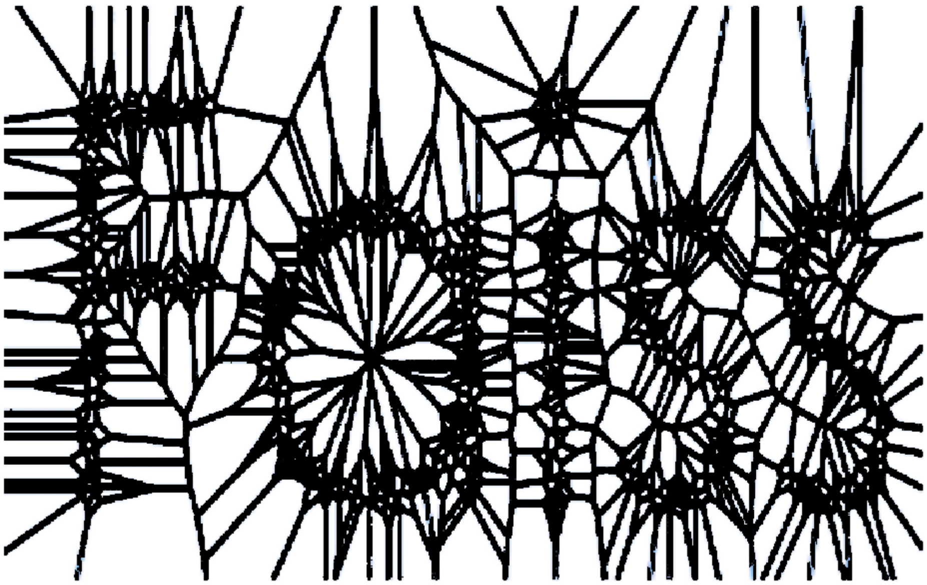
Bringing Intelligence to Storage

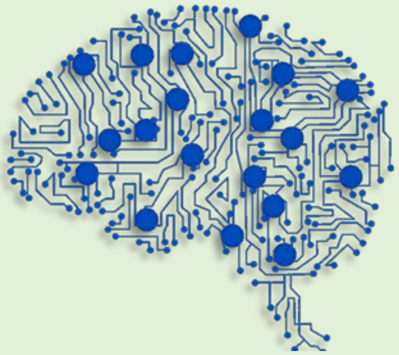


# Turbocharging Artificial Intelligence With NGD Systems Computational Storage



### Key Takeaways

1. Image Similarity Search techniques are working, however, searching PB-Scale image sets in real time is a looming challenge.
2. The current scale out image similarity search techniques in standard servers force users to split up the dataset across a large CPU/GPU server cluster, significantly increasing CAPEX and OPEX.
3. Utilizing NGD Systems Computational Storage allows compute to scale linearly with capacity avoiding long data load times experienced with current server use.
4. **Result**: **In-Situ Processing Constant Performance** vs. *traditional exponential performance degradation*



# The Cost of Scaling Artificial Intelligence

We have all seen demonstrations of the capabilities of Artificial Intelligence (AI)-based imaging applications, from facial recognition to computer vision assisted application platforms. However, scaling these imaging implementations to Petabyte-scale for real-time datasets is problematic because:

1) Traditional databases contain structured tables of symbolic information that is built like a table with each image having a row in the index. The tables are then cross-linked and mapped to added indexes. For performance reasons, these index tables must be maintained in memory.
2) Utilizing traditional implementations for managing datasets requires that data be moved from storage to server memory, after which the applications can analyze the data. Since these datasets are growing to Petabyte scale, the ability to analyze complete datasets in a single server's memory set is all but impossible.

Utilizing brute-force methods on modern imaging databases, which can be in the petabyte-size range, is often both incredibly difficult and enormously expensive. This has forced organizations with extremely large image databases to look for new approaches to image similarity search and more generally to the problem of data storage and analysis.

## The Evolution of Vector-Based Similarity Search

Artificial intelligence (AI) techniques offer new approaches to image similarity search that overcome the challenges of brute-force methods. Neural networks are used to train AI tools known as classifiers. Classifiers generate multi-dimensional vectors that are much more powerful and flexible than simple photographs or classical feature analysis attributes. One popular image classifier today is Google's TensorFlow™, which can provide real-time image categorization.

Classifiers provide a powerful set of tools to improve image similarity searches. However, a new storage technology is needed to save CPU and memory cost and scale solutions for petabyte-scale image sets in anything approaching real time. The Facebook Artificial Intelligence Similarity Search (FAISS) was developed specifically to address image similarity searching of petabyte-scale image databases.

## FAISS: Changing the Image Search Game

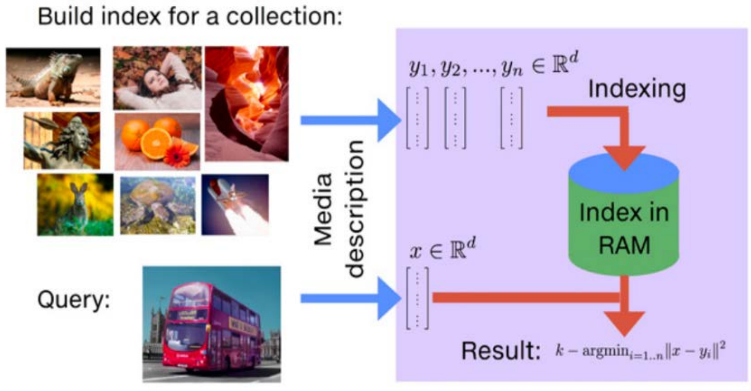

Build index for a collection:
$y_1, y_2, \ldots, y_n \in \mathbb{R}^d$
Indexing
Media description
Index in RAM
$x \in \mathbb{R}^d$
Query:
Result: $k - \mathrm{argmin}_{i=1..n} \|x - y_i\|^2$

**Figure 1: FAISS Workflow**

FAISS provides several capabilities that are unique from other image search approaches. These include: use of multiple similarity search methods with varying tradeoffs; optimization for memory usage and speed; and implementation aimed at traditional HW platforms that require added compute. Figure 1 illustrates the functional flow of the FAISS library. FAISS works to balance three specific metrics: speed (how long it takes to return many similar vectors); memory usage (how much RAM is required); and accuracy (does the list match the brute-force results).

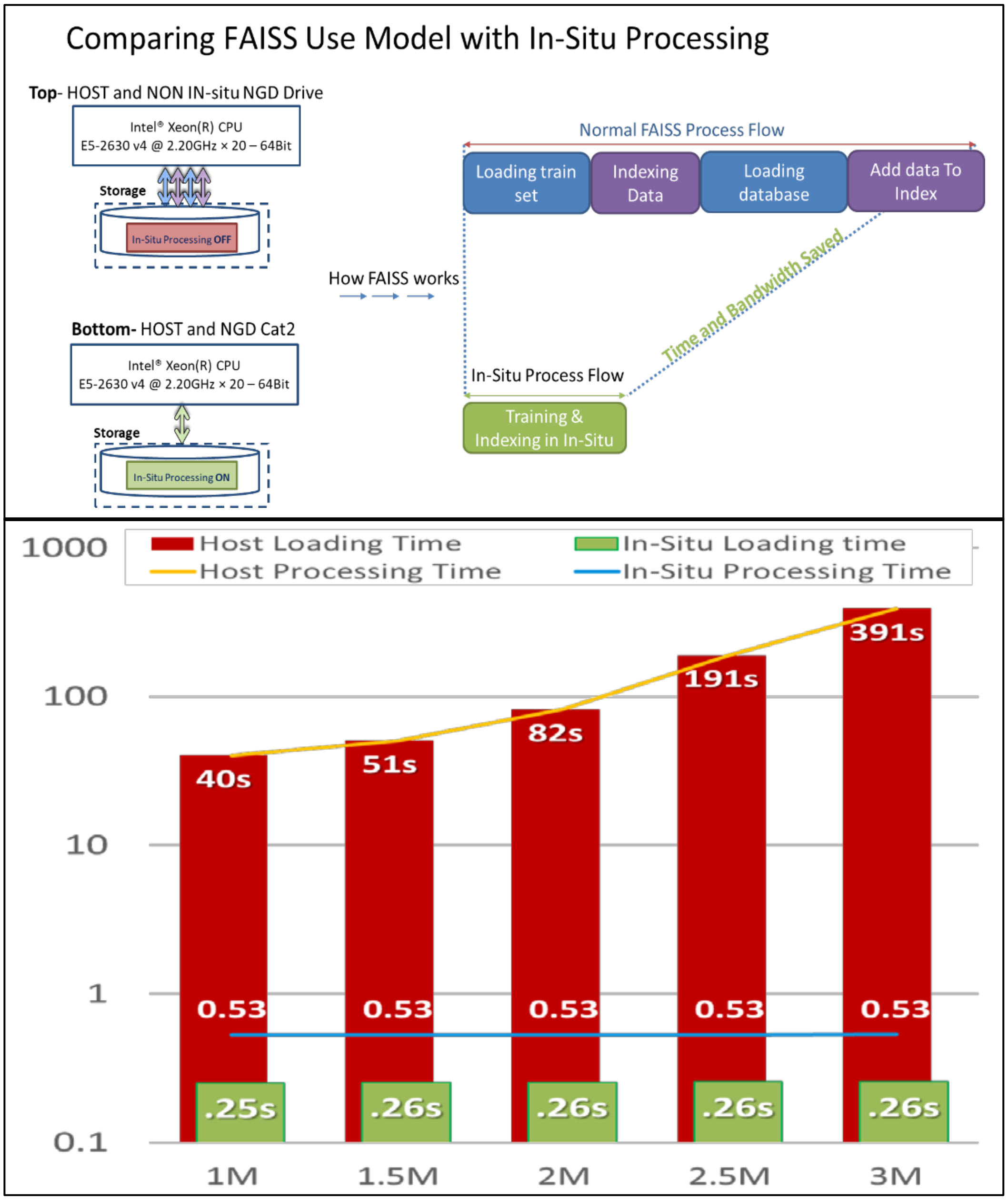
While FAISS and similar approaches streamline storage and processing of large data sets, they don't eliminate the delay inherent in moving petabyte-scale datasets from storage devices into server main memory, or reduce memory usage (which scales linearly with the dataset size). This data movement time and the memory footprint are critical problems that prevent image similarity solutions from scaling and providing real-time results. The current method to address this problem is to split the data sets across multiple servers, even a large server cluster. In these cases, a petabyte of data AND the future PCIe 4.0 bus don't reduce the data set movement and will still take an agonizing amount of time to manage. The obvious (but often overlooked) solution to the data movement problem is simple: Stop Moving the data, let the storage do the work.

## Augmenting FAISS with NGD Systems Computational Storage

This is the exact concept created and implemented by the Computational Storage devices from NGD Systems. By embedding multi-core ARM 64-bit application processors in the storage devices, the query management and the search portions of FAISS can be executed without moving the data out of the storage device. Adapting FAISS to NGD Systems Computational Storage does not require any kernel changes, and only minor code modifications to split off the parts that run in the computational storage. Utilizing NGD's computational storage API, making these changes takes a few weeks, and not months. The NGD Systems host agent then manages the exchange with the storage nodes and mitigates the data traffic.   All of this happens in near real time and with no changes to the block level characteristics of the Computational Storage NVMe devices themselves.

## The Impact of Computational Storage on FAISS Execution Speeds

To test the effectiveness of computational storage using FAISS, NGD Systems utilized the test setup shown. In both cases, the host system is an Intel® Xeon E5-2630 v4 processor operating at 2.2GHz. In the Top configuration without In-Situ Processing, Catalina-2 Storage Devices were utilized in a traditional storage model. In the Bottom configuration, NGD Catalina-2 U.2 Storage Devices were used with the patented In-Situ Processing turned ON. The completion times for the standard server in our test example was over 6 minutes for the data set used. The standard server completion times were ultimately dominated by time the data had to be moved from storage to memory. In addition, the results showcase the effects of the required copy-execute-flush-repeat process in memory, were fewer and less accurate results for optimal use of the tool, requiring more interaction by the user to sort the data.



In contrast, the system where In-Situ Processing was enabled showed consistent and scalable completion times that never exceeded half a second regardless of the dataset size. Moreover, the completion time was achieved while standard NVMe Storage access was maintaining a five-9s QoS for other data. Matching this performance with standard servers would require a larger cluster of servers. There is a significant reduction in overall processing cost when using Computational Storage. Concrete reductions in CapEx (less system memory), as well as ongoing costs associated with OpEx (heating and cooling) can be significantly reduced in large clusters using this technology.

## NGD Systems Computational Storage: Accelerating FAISS

For image analysis to be used in real-time situations, it must be accurate, economical, and fast enough that it does not impede other workflows. Computational storage from NGD Systems provides a critical technology that both accelerates FAISS, and significantly reduces both the CapEx and OpEx needs of the organization. More importantly, it provides a means to scale computational resources linearly with storage needs, and in doing so eliminating the data movement penalty that is inherent in standard server cluster-based image recognition approaches. The result is near real-time performance, completion times that stay constant as dataset size scales up, and significantly reduced CapEx and OpEx. This is why NGD Systems Computational Storage changes the game for data processing and analytics.