Reducing Cost, Increasing Results

Even the world's largest, most advanced hyperscale datacenters are obsolete because they rely primarily on traditional server hardware built around the basic method of using Von Neumann architecture. This 70-year old computer architecture for nearly all general-purpose servers has never experienced the right kind of change. However, recent advancements in storage interface speeds (i.e. NVMe), exponential growth in the size of data sets, and the challenges of storing it all have exposed the limitations of this traditional architecture. Allowing for new ways to implement Von Neumann is ways never tried before.

At the heart of the issue is the rapidly growing cost of moving data within and between servers due to the size (at or beyond the ability to have 32TB per drive) in the data set. The scale-out cluster model of HDFS attempts to address this issue by executing tasks on nodes closest to where the data resides; but data movement between nodes and local compute/memory complexes is still necessary for Big Data, and the costs become significant as clusters increase in size. Luckily, there is a better way!

Doing Data Inventory with Current Architectures

Imagine if your favorite warehouse club store needed to check inventory and wanted to send out-of-season products back to their vendors before the limit expired, but it required moving their entire warehouse of products temporarily to a remote "processing center" several miles away. They would have to:

- Hire additional staff and rent a fleet of container trucks to move the entire stock down the local 2-lane highway to the processing center
- Transport and use the additional staff to unload the stock into the processing center
- Have the processing center manager count the inventory and make any adjustments
- Reload the correct inventory back on to the fleet of trucks and drive it back to the store warehouse
- Finally move all the inventory back into the warehouse, before letting staff go.

To top it off, it must be done in one day! Sounds crazy, doesn't it?

Common sense tells you it is a tremendous waste of time, energy, and resources to move the inventory back and forth. Of course, in real life, each warehouse store would have an inventory manager who performs audits on site and sends just the specific inventory in question back to the vendors. However, even this more accepted method requires one or more people to visibly inspect, record, maintain, and report the results of walking the aisles. This leads to mistakes (no

one is perfect) and if you add redundancy you incur more cost and time. We wonder, what else can be done to improve this process?

This is precisely the situation with traditional compute architecture. Modern x86 computers are based on physicist John von Neumann's design for a digital computing system consisting of a processing unit, volatile memory for data and instructions, external mass storage (non-volatile memory) and input/output mechanisms. The architecture dictates that data must be moved from storage to memory in a typical data access, as follows:

- (1) CPU initiates transaction to read from Storage.
- (2) The data is transferred from Storage to DRAM.
- (3) CPU performs computations on the data and results are moved back to DRAM.
- (4) Processed results data is transferred to the host and then written back to Storage.

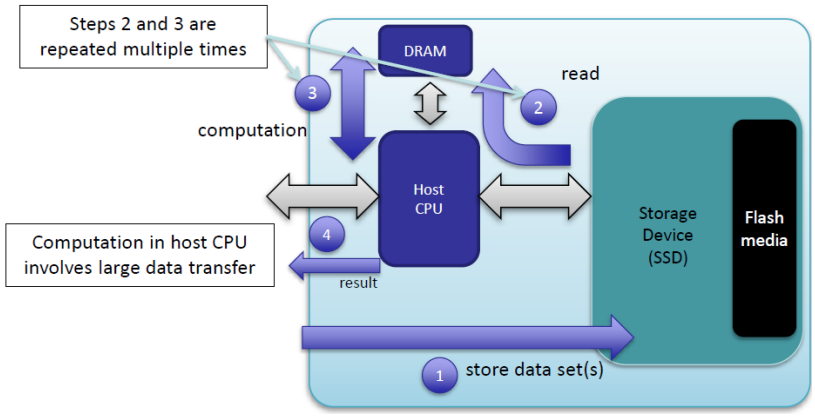

Figure 1 - Typical Compute Architecture Data Access

We know intuitively, that moving massive amounts of data from storage to host CPU memory to process a query is costly in terms of power consumption and time. Until recently, the size of typical data sets has allowed this architecture to work adequately. However, as data sets grow and data-intensive applications such as Big Data analytics, artificial intelligence (AI), machine learning (ML), genomics, and IoT gain in use, the costs and time needed for data movement is becoming critically challenged. This is where distributed processing and allowing more efficient data management by migrating this compute architecture inside the drives themselves.

The Data Movement Costs Challenge Explored

The impact of data movement is being felt in nearly all compute applications. Even in consumer devices such as smartphones, tablets, mobile-PCs, and wearable devices, where cloud services are becoming necessary, it has been shown that data movement between the main memory system and computation units accounts for, on average, 62.7% of the total system energy.[i] This is an important insight into challenges faced by battery powered devices. However, in the datacenter, the energy required to move data has profound implications for scalability. Energy can make up 42% of the total monthly operating cost in data centers[ii], so it is imperative to develop methods to become more data and energy efficient.

How then do we quantify the costs? At a high level, we can look at the costs of moving large quantities of data in terms of the time and energy required. Many industry researchers have

studied these metrics and proposed ways to mitigate data movement for energy savings. As you would expect, measurements of time and energy used vary according to system configuration and use case. The methodologies to calculate energy required in each stage of data transfer and compute are extremely complex and saved for a future discussion, but we can summarize the findings of industry studies showing the significant costs of large-scale data movement.

Other studies have determined the energy by the average time to process a single input byte. Researchers derived the energy per byte by running several data-intensive application kernels, and measuring instructions per byte (IPB), cycles per byte (CPB), and cycles per instruction (CPI). By comparing CPB, researchers can determine relative efficiencies of different compute architectures.[iii] Ultimately, it has been found that, on average, applications spend more than 55% of execution time moving data in certain use cases. [iv]

With applications such as AI and ML, which work with huge amounts of raw data (structured and unstructured) and require high computing power to "learn", the compute capability has become the bottleneck. In the traditional scale-out model, this issue is addressed by adding nodes for more distributed compute power and more memory. Unfortunately, adding server nodes is costly from both CapEx and OpEx perspectives. Adding more nodes also increases the length of interconnects, thereby increasing time required for data movement.

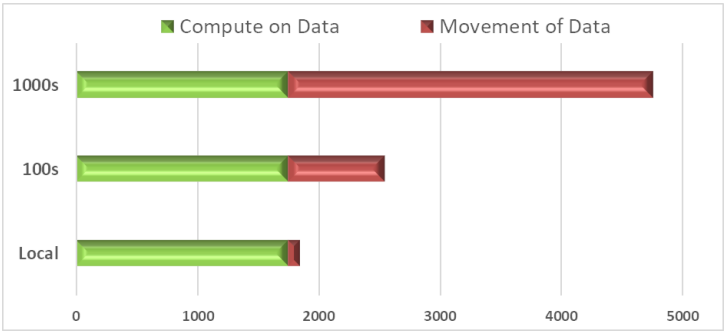

Figure 2 - Energies per byte with different sizes of clusters, I. Choi, *Energy Efficient Scale-In Clusters with In-Storage Processing for Big-Data Analytics*

Moreover, industry studies show that the energy consumption of data movements will dominate the total energy consumption as the number of nodes in the cluster increases (Figure 2).[v] Therefore, the scale-out architecture with traditional commodity servers becomes more inefficient and costly as the dataset grows! Once again, it begs the question - is there a better way?

The Computational Storage Solution

In storage, the best way is by "Bringing Intelligence to Data". This elegant, common-sense solution to minimizing data movement is to bring the compute capabilities to the data – now known as Computational Storage. Implemented in a standard 'off the shelf', high-capacity NVMe SSD, Computational Storage provides a game-changing solution to the out dated Von Neumann compute architecture.

For applications with large data stores and significant search, indexing, or pattern matching workloads, Computational Storage offers much faster results than using valuable host memory and CPU to process the massive data sets. This "In-Situ Processing" enables existing

applications to scale compute resources linearly within the storage, making it possible to manage data sets in place and reduce memory requirements and CPU load.

Data Movement is costing more than just time, but money, resources and wasted analytics. It is time to take the next step in storage and implement Computational Storage NVMe SSDs. Following the steps of the data, noting how much more is done inside the drive itself saving on the data movement time and increasing efficiency and reducing Host CPU and memory loads.

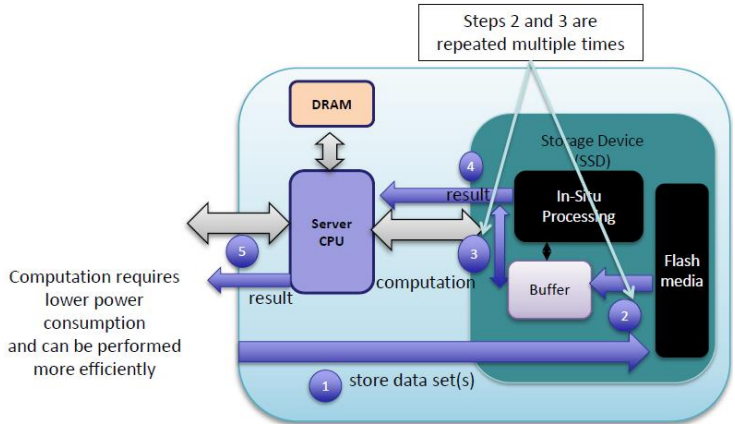

Figure 3 – Data Access with Computational Storage

NGD Systems demonstrates the incredible value of Computational Storage in ground-breaking and industry first POC studies:

- Using its Newport Computational Storage platform in a single server, NGD Systems has demonstrated with Facebook Artificial Intelligence Similarity Search (FAISS), queries per second are significantly improved, resulting in near real-time performance.
- Based on initial analysis, running the National Institutes of Health's (NIH) BLAST tool on a 150TB database, in a single core server with 64 NGD Computational Storage devices, NGD Systems predicts performance comparable to a 32-core server environment.

NGD Systems is the pioneer in design and manufacture of intelligent SSD solutions, leveraging decades of storage and SoC design experience. When it comes to pathfinding, '***Bringing Intelligence to Storage***' is the paradigm shift needed to allow data sets to grow and execution time to reduce!

Contact us at Info@NGDSystems.com to learn how Computational Storage can elevate your datacenter projects while saving Time, Money and Power!

References

[i] A. Boroumand, et.al., *Google Workloads for Consumer Devices: Mitigating Data Movement Bottlenecks*, March 2018
[ii] J. Wang, et.al., *SSD In-Storage Computing for List Intersection*, DaMoN'16, June 2016.
[iii] S. Cho, et.al., *Active disk meets flash: A case for intelligent SSDs*, ICS'13, June 2013
[iv] HW Tseng, et.al., *Gullfoss: Accelerating and Simplifying Data Movement among Heterogeneous Computing and Storage Resources*, UCSD, November 2015.
[v] I. Choi, Y. Kee, *Energy Efficient Scale-In Clusters with In-Storage Processing for Big-Data Analytics*, MEMSYS, October 2015.