



Computational Storage Drive (CSD) and Data Processing Unit (DPU): Foe or Friend?

This white paper discusses the role of the computational storage drive (CSD) and data processing unit (DPS) in data center infrastructure. Data centers prioritize on cost saving, and its hardware TCO (total cost of ownership) can only be reduced via two venues: (1) cheaper hardware manufacturing, and (2) higher hardware utilization. The inevitable slow-down of CMOS technology scaling forces data centers to increasingly rely on the 2nd venue, which naturally leads to the current trend towards compute/storage disaggregation as illustrated in Fig. 1.

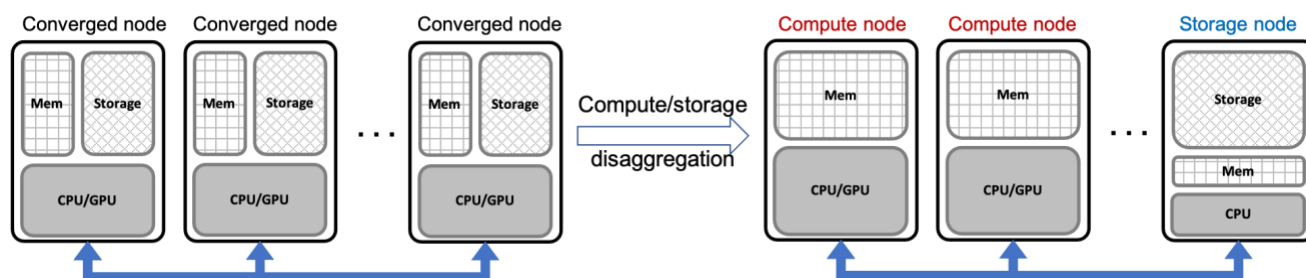


Figure 1: Illustration of the transition towards compute/storage disaggregation in data centers.

Even though not including the term “computation” in its name, storage nodes in disaggregated data centers are responsible for a wide range of heavy-duty computational tasks: (1) *Storage-centric computation*: Cost saving demands the pervasive use of data compression in storage nodes. Meanwhile, storage nodes must ensure at-rest data encryption. Moreover, data deduplication and RAID or erasure coding could also be on the task list of storage nodes. All these storage-centric tasks demand a significant amount of computing power. (2) *Network-traffic-alleviating computation*: Disaggregated infrastructure may further impose a variety of application-level computational tasks on storage nodes in order to greatly alleviate the burden of inter-node network. In particular, compute nodes may off-load and/or replicate certain data processing functions to storage nodes in order to largely reduce the amount of data that must be transferred between compute nodes and storage nodes. Accordingly, disaggregation can fall into two categories as illustrated Fig. 2: (i) *Baseline disaggregation*: storage nodes are primarily responsible for the storage-centric computation, which is applicable to general-purpose systems; (ii) *Advanced disaggregation*: In addition to storage-centric computation, storage nodes must handle a significant amount of network-traffic-alleviating computation for specific applications such as database and data analytics.

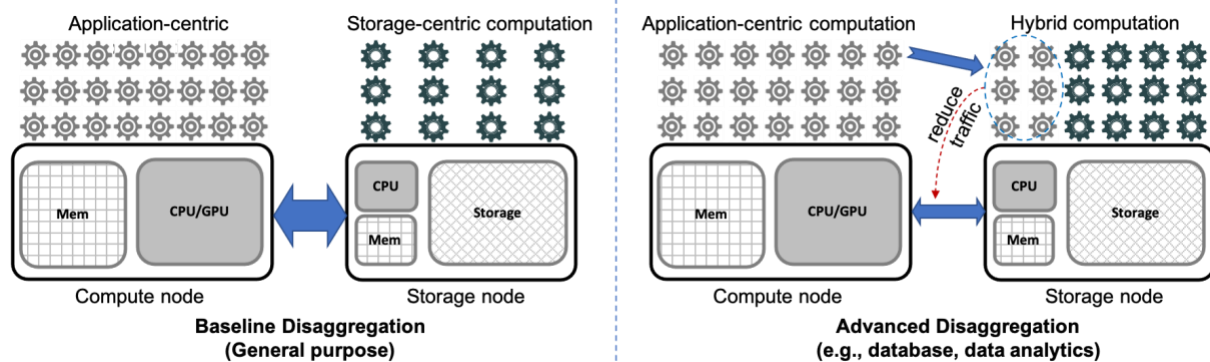


Figure 2: Illustration of two categories of disaggregation that differ on the storage nodes' computation duties.

Basics of DPU and CSD

To reduce the storage node TCO without compromising the computational capability, the most intuitive (and most effective) option is to migrate heavy computational tasks from CPU to customized (hence more cost-efficient) hardware engines inside storage nodes. This intuition has led to rapidly growing interest and investment on the following two types of emerging products:

1. Data processing unit (DPU): The term *DPU* essentially evolves from *network processor*. To avoid overwhelming CPUs with network processing in the presence of ever-increasing network bandwidth (e.g., 10Gbps→40Gbps→100Gbps), data centers have widely deployed SmartNIC (smart network interface card) that uses a dedicated network processor to off-load heavy-duty network processing operations (e.g., packet encapsulation, in-transit data encryption, and more recently NVMe-oF) from host CPU. To further enhance their value proposition, network processor chip vendors lately started to move beyond the network domain into the storage (and even general-purpose) domain. In particular, network processor chips are augmented by integrating additional hardware engines (e.g., compression), more embedded processors (e.g., ARM cores), and stronger PCIe connectivity (e.g., PCIe switch with multiple ports). Accordingly, the term DPU was coined to distinguish from the traditional network processor. As illustrated in Fig. 3, from the functionality's perspective, a SmartNIC can be considered as a subset of a DPU-based card. Meanwhile, different from SmartNIC, a DPU-based card separates host CPU from SSDs and directly controls all the SSDs via a backplane. In disaggregated data centers, storage nodes can utilize DPU to handle storage-centric and network-traffic-alleviating computational tasks, in addition to network processing.

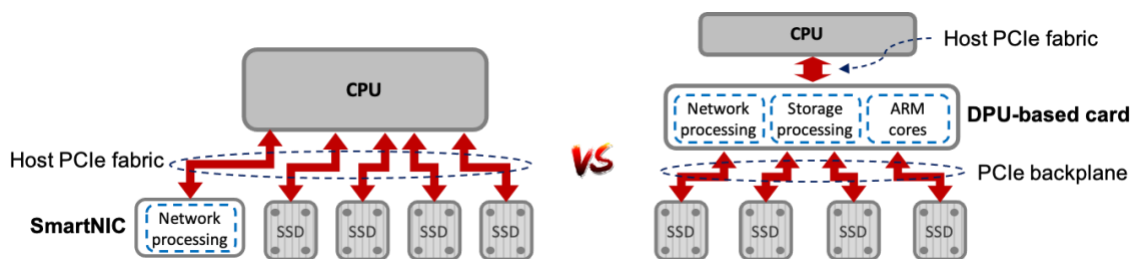


Figure 3: Storage nodes deploying SmartNIC vs. DPU.

2. Computational storage drive (CSD): As the natural product by combining the wisdom of “moving computation to data” and the growing importance of heterogeneous computing, CSD has quickly gained significant momentum and is well poised for its prime time (e.g., there is already an on-going industry-wide effort on expanding the NVMe standard to support CSD). Fig. 4 illustrates the architecture of CSD that can effectively assist storage nodes to handle the storage-centric and network-traffic-alleviating computation. To best serve storage-centric computation, CSD should provide the best-in-class support of compression and encryption, in both in-line and off-load modes.

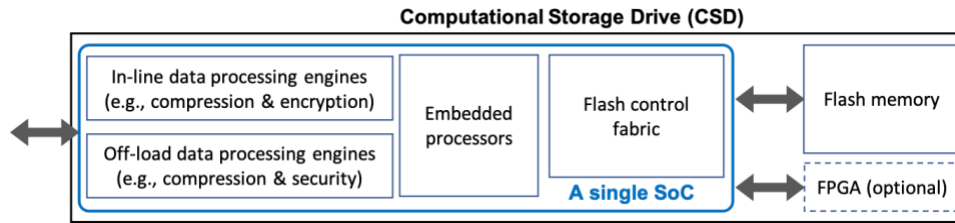


Figure 4: Architecture of CSD geared to storage nodes in data centers.

For the in-line mode, CSD implements data processing such as compression and encryption directly along the storage IO path, being transparent to the host. With the minimal latency overhead, in-line data processing is highly desirable for latency-sensitive applications such as relational database. CSD may integrate additional hardware engines to provide off-loading service through well-defined APIs. In addition to relieving CPUs from compression/encryption tasks, they make it possible for CSD to carry out in-storage processing over data that are compressed/encrypted at the application level.

Regarding network-traffic-alleviating computational tasks, their application-dependent nature demands a programmable computing fabric inside CSD. The most convenient choice is of course to integrate embedded processors (e.g., ARM cores). It is also possible for CSD to integrate an optional FPGA (field-programmable gate array) device to further strengthen the programmable computing power. It is evident that “computation” and “storage” inside CSD must cohesively and seamlessly work together in order to provide the best possible end-to-end computational storage service. In the presence of continuous improvement of host-side PCIe and memory bandwidth, tight integration of “computation” and “storage” becomes even more important for CSD to not become the throughput bottleneck. Therefore, it is necessary to integrate “computing fabric” and “flash control fabric” into one SoC (system-on-chip) as illustrated in Fig. 4.

After the above brief introduction to DPU and CSD, let us next study how storage nodes in disaggregated data centers could utilize DPU and/or CSD. Following the categorization as illustrated in Fig. 2, we will discuss the baseline disaggregation and advanced disaggregation separately.

DPU and CSD: Competitor in Baseline Disaggregation

In the case of general-purpose baseline disaggregation, storage nodes are primarily responsible for storage-centric computation (in particular compression and encryption). Fig. 5 shows the compression and encryption flow: An appropriate compression block size (e.g., 4KB or 16KB) is chosen in adaptation to the data access characteristics and latency constraint. Due to the runtime data compressibility variation, the compressed block size (largely) differs from one another. Encryption is done in a much finer granularity, e.g., AES (advanced encryption standard) operates in the unit of 16B block. Accordingly, one must manage the mapping between each raw (uncompressed) data block and its compressed/encrypted data block on the storage device, which is realized by employing sophisticated data structure and management algorithms (e.g., log-structured data management).

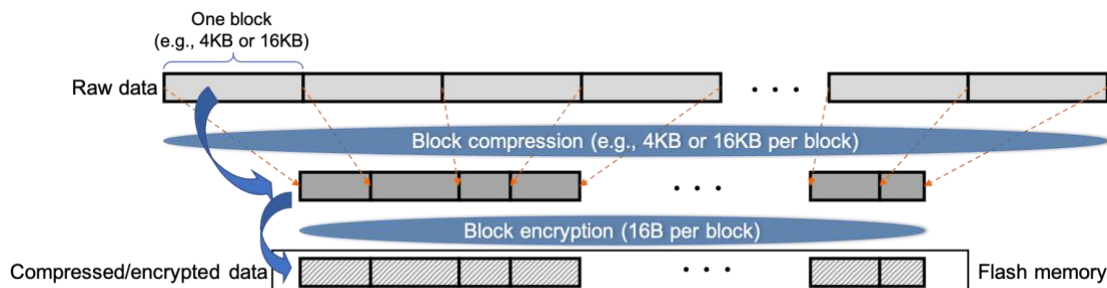


Figure 5: Illustration of data compression and encryption flow.

Fig. 6 illustrates the data flow when storage nodes deploy DPU to handle compression and encryption. In this context, DPU itself must manage the raw-data-to-compressed-data mapping across all the underlying SSDs. Meanwhile, DPU must carry out compression/encryption on all the data being sent to and fetched from all the SSDs. Since the single DPU chip handles all the data processing on the network and storage path, which could lead to potential interference between different data flows. To accommodate NAND flash memory operational characteristics, each SSD internally implements a flash translation layer (FTL) to manage data mapping on flash memory, as illustrated in Fig. 6.

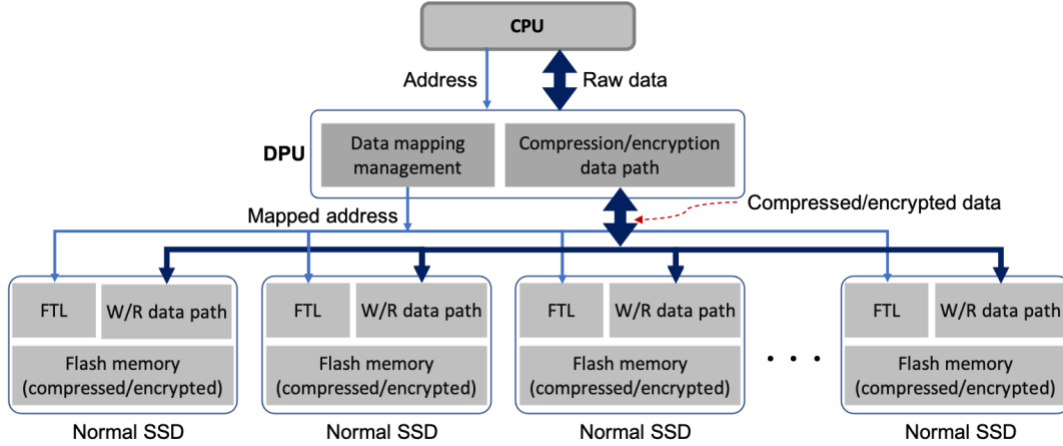


Figure 6: Illustration of data flow when using DPU to handle compression/encryption across all the SSDs.

Fig. 7 illustrates the data flow when storage nodes replace commodity SSDs with CSDs and distribute compression/encryption into all the CSDs. Each CSD internally implements enhanced FTL (eFTL) that merges the compressed data block mapping management layer into conventional FTL to form a lean and unified management firmware layer. Meanwhile, each CSD internally implements compression/encryption engines that are tightly integrated with the flash memory write/read data path.

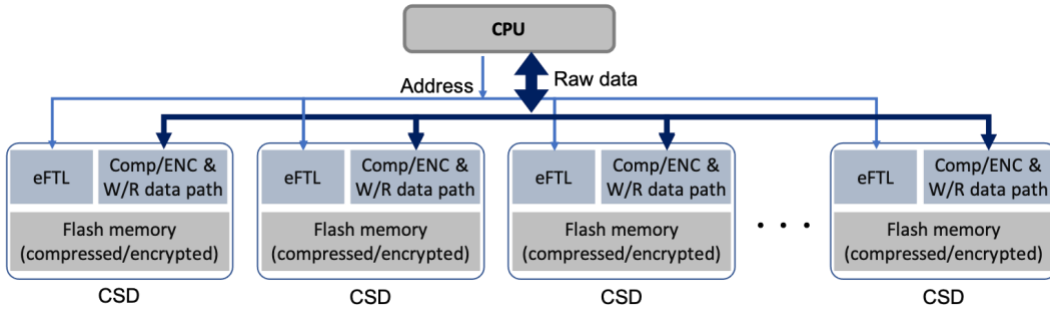


Figure 7: Illustration of data flow when distributing compression/encryption into CSDs.

Apparently, DPU and CSD are *competing options* for storage nodes to relieve CPUs from heavy-duty storage-centric computation (in particular compression and encryption). Based on the above description and comparison as illustrated in Fig. 6 and Fig. 7, one could easily observe that CSD is a much better option mainly because:

1. *Native scalability*: The DPU-based option centralizes all the compression/encryption computation into the single DPU chip. The computing power does not scale with the installed storage capacity (i.e., the number of SSDs), and the central DPU could potentially become the hot-spot and hence throttle the entire system. In comparison, by distributing the compression/encryption computation across all the CSDs, the CSD-based option natively scales the computing power with the installed storage capacity, and fundamentally eliminates potential hot-spot in the system.
2. *Shorter read latency*: The DPU-based option is subject to longer data access latency for two issues: (i) All the data block decryption/decompression operations are queued in the single DPU chip, which could add

- noticeable read latency overhead; (ii) The extra data mapping management inside DPU could incur extra read latency overhead. Clearly, these two issues disappear when CSDs are being deployed in storage nodes.
3. *Seamless deployment*: Storage nodes can simply replace commodity SSDs with CSDs and keep existing SmartNIC, without making any additional changes to their hardware/software. As a result, storage nodes can upgrade CSD and SmartNIC separately in response to the independent evolution of storage and network hardware. In comparison, DPU must directly control SSDs through a separate backplane, which could demand hardware customization in storage nodes and meanwhile largely under-utilize the PCIe connectivity of host CPU. Moreover, storage nodes must upgrade DPU in order to upgrade either storage or network hardware, leading to potentially higher system maintenance/upgrade cost.

DPU and CSD: Co-worker in Advanced Disaggregation

As discussed above, in the case of advanced disaggregation, storage nodes are also responsible for additional application-level computation (in particular network-traffic-alleviating computation) in order to further reduce the entire data center TCO. In this context, DPU and CSD could be complementary to each other and collectively form a low-cost, high-performance heterogeneous and distributed platform. As illustrated in Fig. 8, such a platform can most effectively relieve host CPU from the network processing, storage-centric computation, and network-traffic-alleviating computation. The bulk of storage-centric computation (in particular compression and encryption) is distributed across all the CSDs. Regarding network-traffic-alleviating computation, CSDs can only perform relatively simple and localized processing for two main reasons: (i) Within each storage node, data are striped across all the CSDs, which prevents each CSD from performing computations that must be indivisibly carried out over a large continuous range of data; (ii) Each CSD has relatively (much) weaker programmable computing power than the central DPU. Meanwhile, compared with CSD, it is more viable and economic for DPU to integrate special-purpose computing engines that are optimized for assisting network-traffic-alleviating computation. Therefore, as illustrated in Fig. 8, each specific network-traffic-alleviating computation task should be decomposed into two tiers: The bottom tier-1 consists of localized data processing and is distributed to all the CSDs, and the DPU-based tier-2 processing is responsible for handling global data processing.

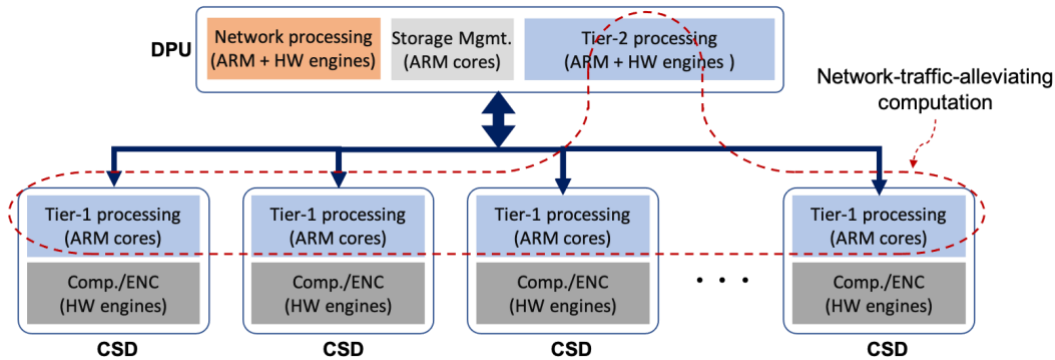


Figure 8: Illustration of DPU-CSD heterogeneous and distributed computing platform.

When storage nodes deploy such a heterogeneous and distributed computing platform to complement with the host CPU, one key issue is how to fully utilize the computing resources in DPU and CSDs for the network-traffic-alleviating computation. Their application-specific nature actually opens a wide range of innovation opportunities, which remain largely unexplored today. We expect that future research should focus on the following two aspects:

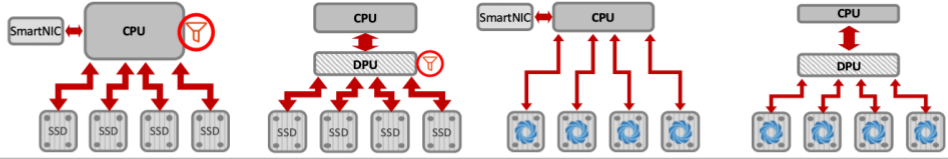
1. *Heterogeneity-driven computation reformulation*: For any given computational task, we should investigate how one could reformulate the computation model/structure in order to better fit the underlying heterogeneous and distributed computing platform. A large portion of the reformulated computation should consist of localized data processing over small data blocks, aligned with the data stripping across all the CSDs. The CSD-based computation should be streaming in nature, do not involve complex control, and do

not demand heavy data buffering/caching. Moreover, under the reformulated computation model, there should be minimal or no data processing iteration between the DPU and CSDs.

2. *Heterogeneity-driven data model tuning*: Most applications construct and configure their data model (e.g., column-store) explicitly or implicitly assuming homogeneous data processing, which could be ill-fit to our target heterogeneous and distributed computing platform and make it difficult to appropriately reformulate the computation. Therefore, we should study how the data model could be accordingly tuned (or even modified) to facilitate the computation reformulation.

Conclusion

This white paper discusses the role of the emerging DPU and CSD products for storage nodes in disaggregated data centers. For baseline disaggregation where storage nodes are primarily responsible for storage-centric computation, either DPU or CSD can be used to reduce the storage node TCO, where CSD is a more scalable and performant option than DPU. For advanced disaggregation where additional application-level computational tasks are imposed onto storage nodes in order to reduce the inter-node network traffic, DPU and CSD could work together to form a heterogeneous and distributed computing platform that can best serve storage nodes. Table 1 summarizes the comparison among different options. At this early development stage of both DPU and CSD, there is still a long journey for the industry to explore their full potential and synergies. We hope this white paper could shed light on the design and implementation of future DPU and CSD products.



	CPU-Only	CPU-DPU	CPU-CSD	CPU-DPU/CSD (adv. disaggregation)
Reduced storage node TCO	✗✗✗	✓	✓	✓✓
Performance scales with capacity	✗✗✗	✗	✓	✓
Energy efficiency	✗✗✗	✓	✓	✓✓
Zero latency overhead	✗✗✗	✗	✓	✓
No data flow interference	✗	✗	✓	✓
No additional backplane	✓	✗	✓	✗
Independent upgrade of network and storage hardware	✓	✗	✓	✗

Table 1: Comparison among different options on the design of storage nodes in data centers.