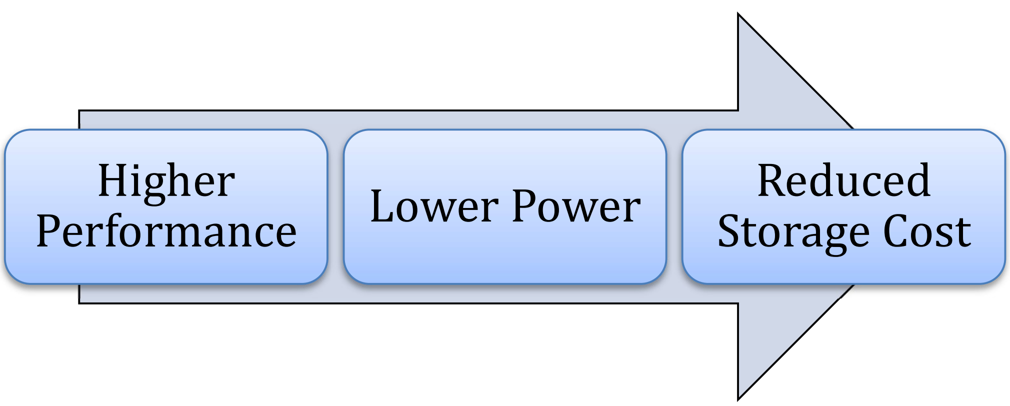
Transparent Compression using NoLoad® Computational Storage Processor

Eideticom's NoLoad achieves higher Compression Ratio and Throughput than software while using 70% less CPU

Executive Summary

- Eideticom's NoLoad Computational Storage Processor (CSP) integrates directly into Linux kernel via a stacked filesystem (NoLoad FS) and provides transparent compression.
- NoLoad enables Reduced Storage Cost (90% lower total cost) by increasing storage capacity, reducing server count, optimizing I/O and maximizing storage lifetime.
- NoLoad provides higher performance by transparently offloading the host CPU (70% less CPU utilized) from compute intensive compression operations, providing vastly improved application performance and superior Quality of Service (QoS).
- NoLoad drives best-in-class power efficiency (60% lower power) as NoLoad compression is dramatically more CPU efficient than software alternatives (gzip-9 and lz4-1).
- Eideticom's NoLoad CSP provides hardware-based compression that enables increased capacity without sacrificing performance.



1 Introduction

In this white paper, we discuss how Eideticom's NoLoad Computational Storage Processor (CSP) can be used to build very performant and cost-effective systems. We outline how to tie an NVMe-based CSP into the software of the Linux filesystem and the benefits of such a tie-in by performing product trials on various storage servers.

Big-Data, High Performance Computing (HPC) and many other applications rely on an underlying filesystem to store data and to retrieve it for analysis in a performant manner. Since the compute servers in these systems can quickly generate gigabytes of data a second, the filesystems used must be both scalable and performant. Failure to do so can result in bottlenecks that do not allow the system to operate at its full potential.

2 NoLoad CSP: An NVMe Computational Storage Processor

NoLoad® is an NVMe-based Computational Storage Processor developed and sold by Eideticom. The NoLoad CSP takes the form of an FPGA accelerator card that performs a variety of storage-centric offload functions while presenting an NVMe compliant PCIe interface to the host. As such, we developed a NoLoad software framework to allow applications, such as Big-Data applications and filesystems, to offload key storage tasks to the NoLoad. This offloading leads to improved performance and efficiency and reduced costs for the storage system.

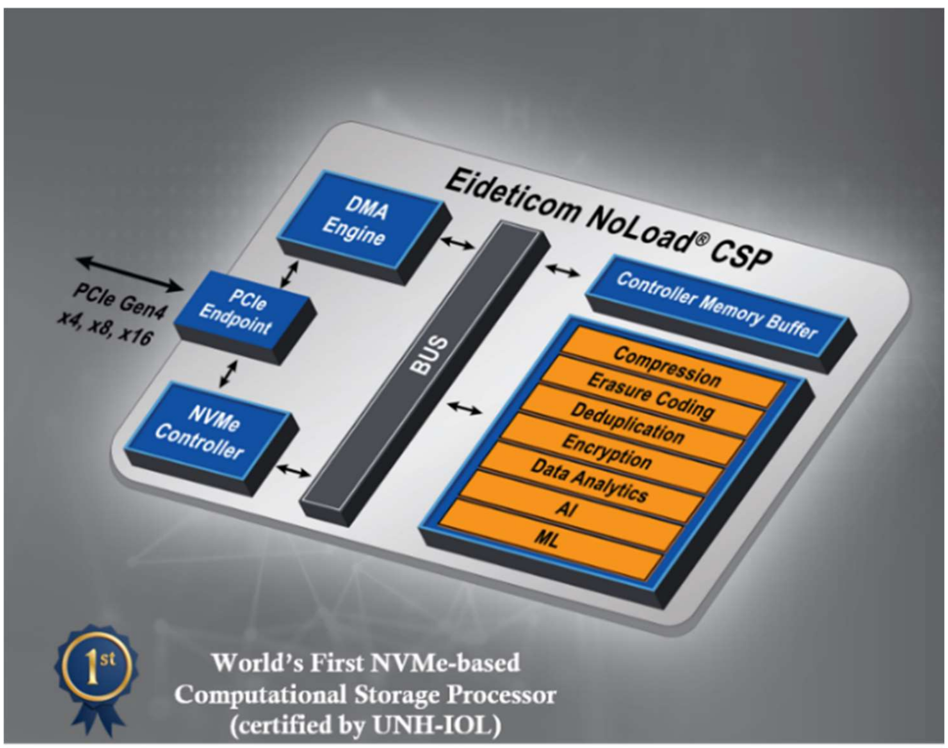


Figure 1: Eideticom's NoLoad Computational Storage Processor (CSP)

3 NoLoad Overview

The NoLoad CSP is purpose-built for the acceleration of storage and compute-intensive workloads**.** NoLoad eliminates performance bottlenecks by embedding processing power directly into the storage fabric. Eideticom's NoLoad supports several acceleration functions, ranging from Compression, Encryption, Erasure Coding, Deduplication and Data Analytics.

NoLoad presents acceleration functions to a host via a standard-compliant NVM Express PCIe interface. There are several reasons Eideticom chose an NVMe interface instead of a proprietary PCIe interface. These include:

- The NVMe driver is inbox in all major operating systems and the Linux NVMe driver is tuned for latency, performance and efficiency. Competing accelerator solutions utilize custom drivers which require porting or may not even work under all operating systems.
- NoLoad's NVMe compliant interface provides seamless integration for all CPU platforms and has been validated on Intel, ARM and IBM Power8/9 CPUs and at PCIe Gen 3 and Gen 4 rates.
- By choosing NVMe, we can leverage a rich ecosystem of open-source tools. In addition to drivers, this includes user-space management tools (e.g. nvme-cli) and user-space frameworks like SPDK.
- Aligning with NVMe allows us to implement and deploy on any PCIe enabled FPGA card and use a common software stack across all form factors.
- We wanted to align with a performant storage protocol as part of the SNIA efforts around Computational Storage. NVMe is the perfect choice for this.
- NVMe has a rich management specification we leverage to manage NoLoad at scale.

4 NoLoad Hardware

Eideticom's NoLoad Computational Storage Solutions are deployed on a wide varity of PCIe-enabled FPGA cards and form factors.

- The BittWare IA-220-U2 are PCIe Gen4x4 FPGA Accelerator Cards developed by BittWare and integrates Eideticom's NoLoad Computational Storage technology. This solution provides Computational Services in a 2.5-inch (U.2) form factor. A typical 2 Rack Unit (2RU) server accommodates 24 NoLoad-U2s.
- The BittWare IA-440 is a PCIe Gen4x16 FPGA Accelerator card developed by BittWare and integrates Eideticom's NoLoad Computational Storage technology. This solution provides Computational Services in a Add-In-Card (AIC) form factor. A typical 2 Rack Unit (2RU) server accommodates two NoLoad-AICs.

Figure 2: NoLoad enabled Bittware IA-220-U2 and IA-440

5 NoLoad Software

Eideticom's rich suite of software enables the connection between acceleration functions on the NoLoad FPGA cards and applications running on a host CPU. Eideticom provide end-to-end computational storage solutions to unlock the full potential of your flash-based storage investmen.

In Figure 3, we showcase the different software components Eideticom has developed. We discuss some of these in more detail:

- **libnoload**. A user-space library that allows applications to leverage the computational storage services offered by NoLoad. Note, that applications need to be updated to access the libnoload API before being offloaded.
- **NoLoad FS**. A Linux-based stacked filesystem that can offload key computation tasks onto NoLoad devices. Eideticom software extends the NVM Express driver to comprehend computation as well as storage.

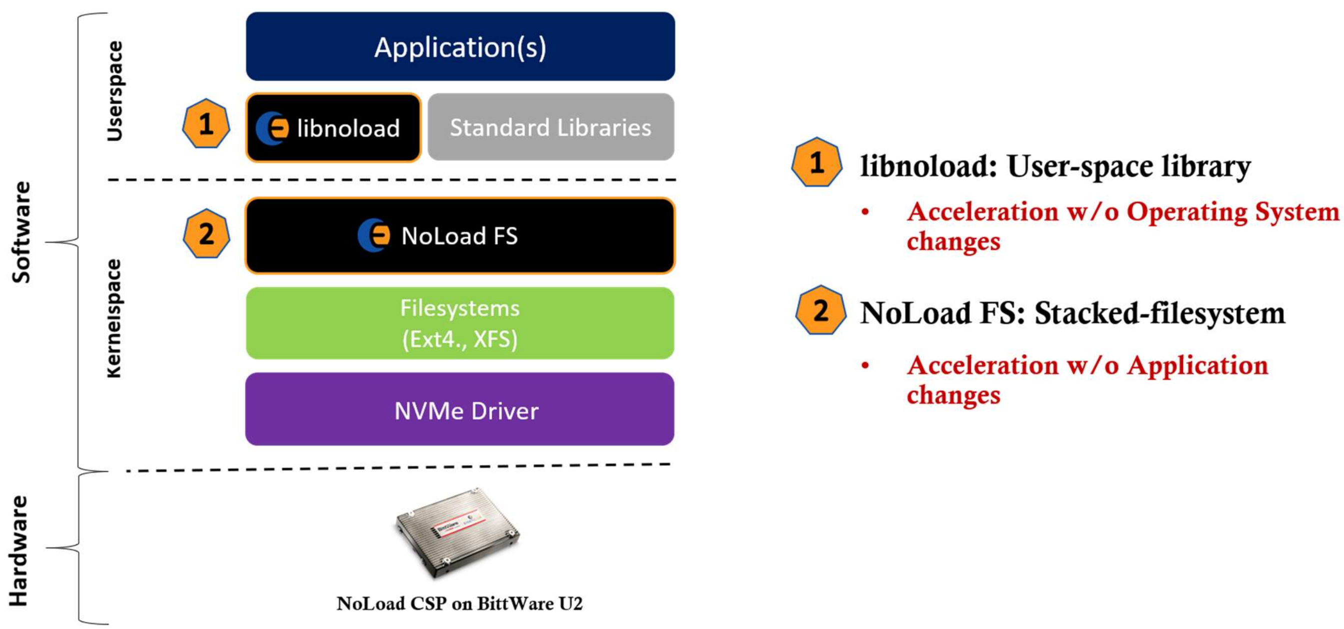


Figure 3: NoLoad Software Stack

6 NoLoad Transparent Compression

In this white paper, we are most interested in offloading compression and decompression to improve the performance and cost-efficiency of a Big-Data system. As such, it is useful to measure the raw NoLoad compression and decompression performance outside of NLFS to establish baseline performance. In Figure 4, we present compression and decompression results for several datasets associated with different application spaces.

For each algorithm, we present the Compression Ratio (CR), the compression input throughput per CPU core (MB/s/core), and the decompression output throughput per CPU core (MB/s/core). We present results for NoLoad compression, a high throughput software compression algorithm (lz4 level 1), and a high-compression software algorithm (zlib level 9). **A single IA-220-U2 can achieve 55Gbps compression ingress and it is anticipated the IA-440 will achieve 100Gbps compression ingress.**

NoLoad performance scales linearly with the number of NoLoad devices in the system. Additionally, NoLoad-U2s and NoLoad-AICs can be seamlessly mixed in the same system to achieve the expected combined throughput performance. The Eideticom NoLoad software stack discussed in Section 5 is capable of discovering and utilizing all the NoLoad devices in a system even when the devices are deployed on different hardware FPGA cards.

Regardless of form-factor, the NoLoad NVMe-based CSP always presents an NVMe interface to the host and the software consumption model remains the same.

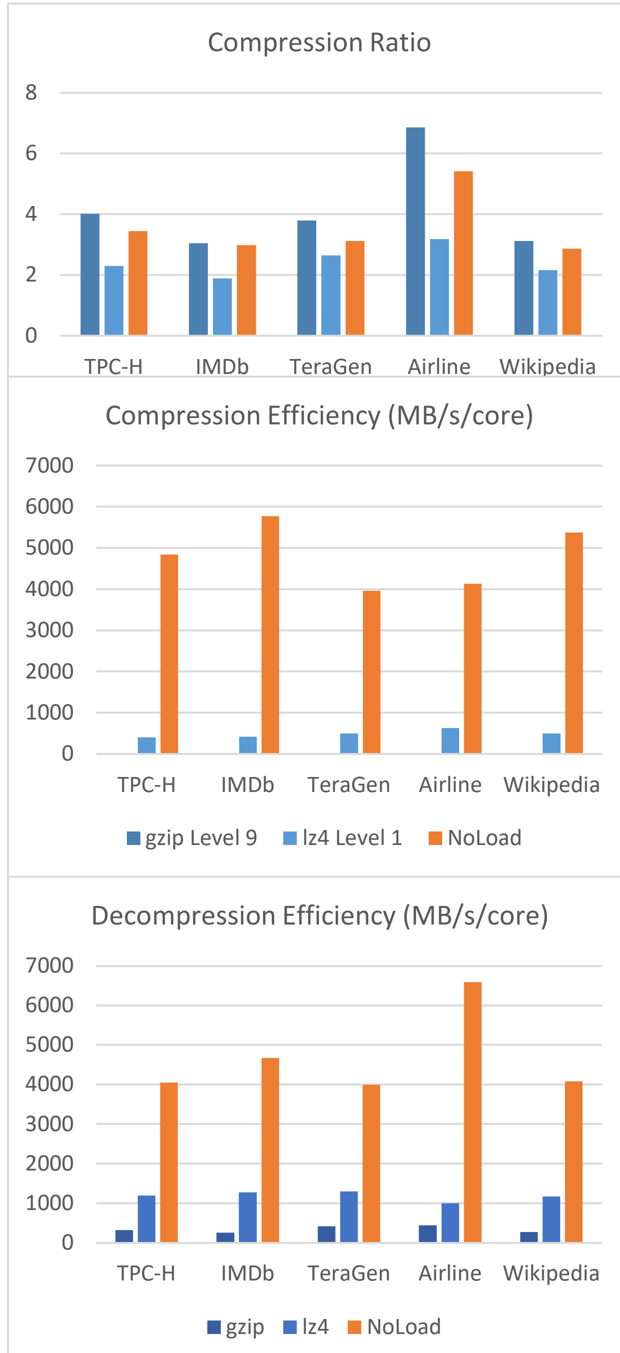


Figure 4: Compression Ratio, Compression and Decompression Efficiency of NoLoad vs. lz4-1 and gzip-9

6.1 Compression and Decompression Conclusions

The main conclusion from Figure 4 is that NoLoad-based compression can provide significant Compression Ratio (CR) advantages over lz4-1, while achieving 8-10 times better efficiency for compression and 4-7 times better efficiency for decompression.

NoLoad's compression ratio is comparable to that of gzip-9 but achieves CPU efficiencies that are orders of magnitude better than software gzip-9. It is this combination of improvements in compression ratio and efficiency (MB/s/core) that makes NoLoad attractive to customers for Big-Data, High Performance Computing (HPC) and many other applications.

6.2 NoLoad Transparent Compression on NoLoad FS

To test the performance gains of a NoLoad-based filesystem we collected data on a number of different configurations. The NoLoad Filesystem (NoLoad FS), discussed in Section 5, was installed on these servers, and Big-Data (TeraGen) data was written to NVMe SSDs with no compression and with compression, both software and NoLoad, via NoLoad FS. The results achieved use two NoLoad IA-220-U2. The main conclusion from Figure 5 is that NoLoad-based compression can provide **significant Compression Ratio (CR) advantages over lz4-1 in Big-Data applications, while also achieving significally better than line-rate throughput.**

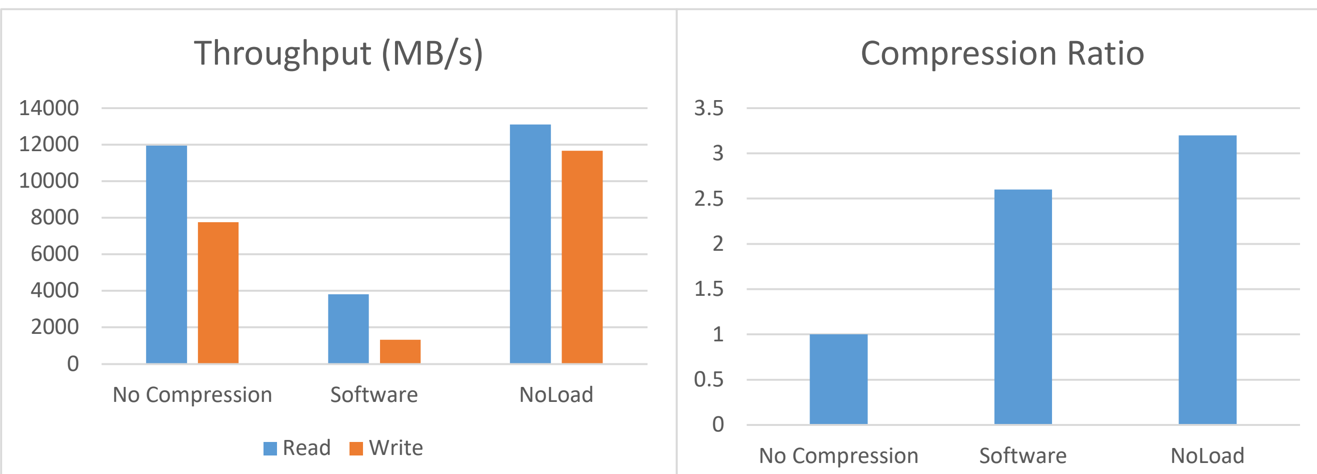


Figure 5: Compression Ratio and Throughput for Big-Data (TeraGen) Data in NoLoad FS

Bottomline - The NoLoad Transparent Compression solution requires no application changes, ties directly into any File System and achieves line-rate compression with high compression ratios.

7 Cost-Benefit Analysis

When performing the cost-benefit analysis of the NoLoad enabled Big-Data application, we considered four key factors, namely:

1. **Storage Capacity Costs:** NoLoad Compression reduces the effective $/GB of the system by increasing the amount of data stored on SSDs. The performance data in Figure 5 demonstrates that NoLoad Compression Ratio (CR) is always higher than lz4-1, therefore increasing the storage capacity of the SSD.
2. **Storage Lifetime Costs:** NoLoad's higher Compression Ratio extends the lifetime of an SSD, since for a given compression input throughput it reduces the Drive Writes Per Day (DWPD).
3. **Throughput Performance Costs:** The performance data in Figure 4 clearly shows that NoLoad compression is 3-6 more CPU efficient than lz4-1 and over 100 times more CPU efficient than gzip-9. NoLoad permits a smaller CPU with less cores to be used to achieve the same throughput. As a result, this leads to fewer servers in the final system, see Figure 6.
4. **Storage Performance Costs:** As noted in the Storage Lifetime Costs, increased CR means the reduced write throughput to the SSDs. NoLoad allows for the use of cheaper, less performant SSDs and that the SSDs will consume less power.
5. **Power Consumption:** The data in Figure 4 demonstrates that NoLoad compression is dramatically more CPU efficient than both lz4-1 and gzip-9, therefore allowing a given CPU to run cooler and consume less power since less cores are fully loaded. NoLoad results in a power consumption reduction of up to 50% per server.

8 Bottomline

Combining the savings from the NoLoad based-system due to Storage Capacity Cost, Storage Lifetime Costs, Throughput Performance Costs, Storage Performance Costs and Power Consumption leads to combined savings of almost **90% per server!**

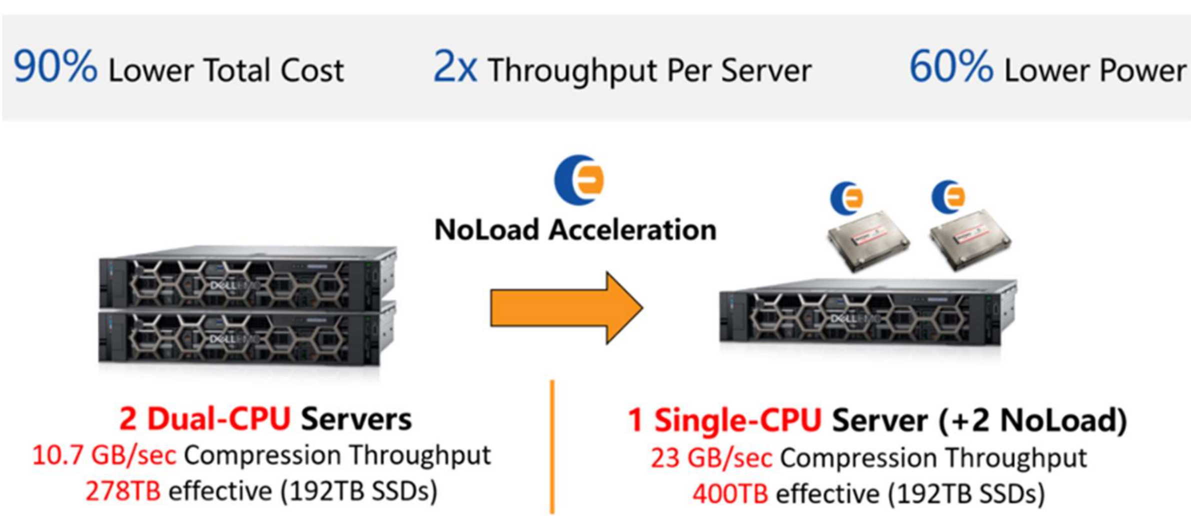


Figure 6: NoLoad Cost-Benefit