MIDDLE CANYON

Klondike:
ScaleFlux CSS 1000
Performance Benchmark

MIDDLE CANYON

Contents

ScaleFlux CSS 1000 Overview

Source: ***"CSS 1000 Computational Storage Subsystem™ Product Specification for 2.x Software Rev 1.4, June 2018"***

The content in this section is copied from the sourced document/site as a convenience to the reader.

The ScaleFlux™ CSS 1000 is a converged PCIe SSD and compute off-load platform powered by flexible, host based management software and reconfigurable FPGA-based hardware. The CSS 1000 is exposed to applications as a standard block device. ScaleFlux software manages the FTL and NVM algorithms such as garbage collection, wear leveling, and error handling. The FPGA provides LDPC error correction and supports compute engines that can be used to improve application performance through CPU offloading and reduced data transfers between the host and storage. A capacitor bank is available to ensure that any committed, in-flight IOs are safely stored to NVM in case of an unexpected power loss.

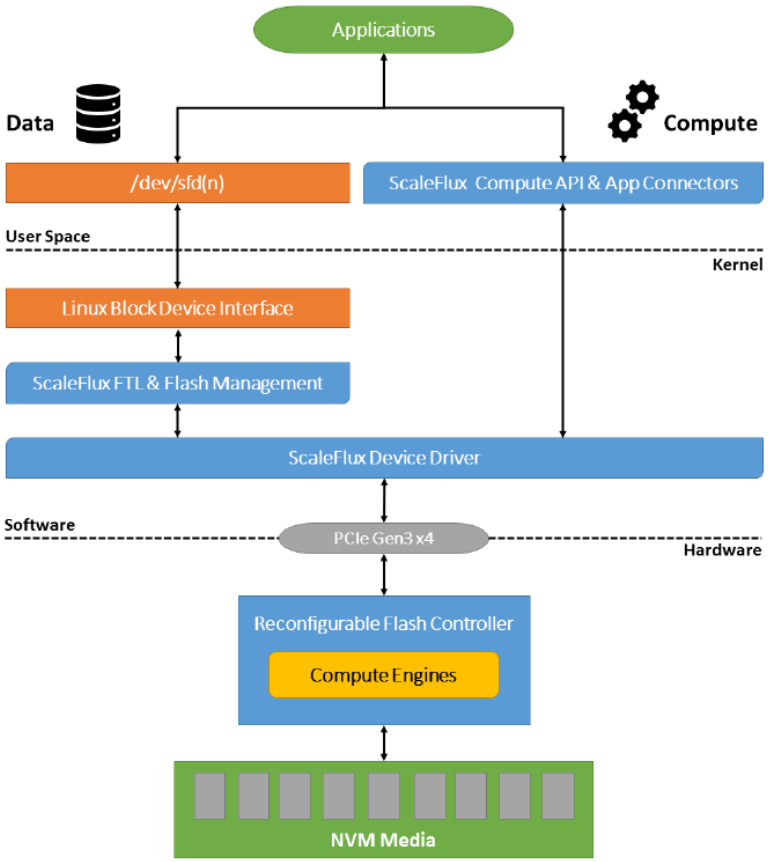


Figure 1: CSS 1000 Block Diagram

Performance

- Up to 725 kIOPS 4KiB Random Read
- Up to 200 kIOPS Sustained 4KiB Random Write
- Up to 3000 MB/s Sequential Read
- Up to 2400 MB/s Sustained Sequential Write

The drivers for the CSS 1000 were provided by ScaleFlux. The drivers are compiled during install based on the system kernel.

Storidge's CIO

Source: http://storidge.com/docs/

The content in this section is copied from the "ezFIO user Guide.pdf" Section 5 as a convenience to the reader. The ezFIO command save all the results to a spreadsheet.

Storidge's CIO (container I/O) software was created to simplify the life of developers, DevOps and storage administrators. Purpose built to remove the pain of managing storage for applications, we make storage simple so you can focus on applications instead of managing infrastructure.

CIO is a clustered block storage solution which is tightly integrated with container orchestrators. It provides a persistent storage layer from which applications can programmatically consume block, file and object storage services. In an orchestrated environment, the data volumes are hyper-converged and exposed on the same host where the application container is running. When a container is rescheduled to another host, the CIO storage orchestrator automatically moves the volume to the new host.

For organizations that want to future-proof their investments in container technology, CIO also supports accessing data volumes from outside the storage cluster. This enables applications running on virtual or physical servers to access the same volumes as the hyper-converged containers.

Storidge technology:

- Rapidly provisions in seconds based on application intent. Profiles minimizes the need for operations to actively manage storage for applications.
- Provisions and manages storage at a container-granular level
- Provides programmatic approach to provision storage through the scheduler and orchestration software
- Orchestrates storage volumes to provide data persistence and high availability
- Ensures redundancy of data by repairing drive and node failures when needed
- Provides performance isolation with quality of service guarantees per containerized application

The Storidge "cio" software version installed was cio-2670-u16.amd64. The u16 is for Ubuntu 16 operating system. The software was provided by Storidge.

Benchmarking Tool: ezFIO

Source: https://github.com/earlephilhower/ezfio

The content in this section is copied from the "ezFIO user Guide.pdf" Section 1 as a convenience to the reader.

Overview

ezFIO is a script-based tool optimized for demonstrating the sustained performance of NVMe storage devices over a series of varying, enterprise-class workloads. It wraps the cross-platform open-source tool FIO (https://github.com/axboe/fio) within an easy to use GUI (Windows) or CLI (Windows and Linux).

Enterprise Workloads

The tested workloads approximate typical use cases for NVMe storage in enterprise servers. As opposed to client workloads, which often focus on interactive performance and have low parallelism, enterprise workloads have higher parallelism and sustained performance needs; however, enterprise workloads also depend on performance consistency (the variability in IO

service times) to meet end-user requirements. Examples of enterprise-class applications would include large scale SQL and NOSQL databases, business intelligence suites, and HPC workloads.

Why Sustained Mode?

The performance of most NVMe device can vary due to the underlying technology and the storage methodology used to write data to the media. For example, NAND flash devices require the implementation of garbage collection to manage data operations on flash chips that do not allow for the updating of existing data. Also, preconditioning or seasoning is a method often employed to ensure that the devices under test are fully utilized and will require garbage collection to operate. This simulates, as closely as possible, the long term performance of these devices better than simple, immediate testing.

The preconditioning operations can require substantial time to complete while the write performance, size, and current state of the device are tested. Preconditioning stages can require some 15 minutes on smaller, faster devices, but up to several hours on larger, multi-TB flash devices.

Why use FIO?

ezFIO is designed to track sustained performance by performing a series of scripted tests that include random and sequential preconditioning. It also operates at the lowest level possible on the raw device to ensure that file system overhead is removed and that the true device characteristics can be obtained. FIO was chosen because of its power and cross-platform capabilities; it can run on everything from embedded ARM Linux systems to the latest Microsoft server and desktop operating systems. Because FIO is so powerful it is also often difficult to configure and run properly. ezFIO implements a predefined sequence of operations and manages the selection of the appropriate command line options. Also, because many users are unfamiliar with the concept of sustained performance, they do not properly precondition the NVMe drives prior to running FIO to obtain realistic, long-term performance rates, resulting in misleading outcomes that do not reflect real-world performance.

Making Test Results Easy to Use

ezFIO will generate a simple OpenDocument Spreadsheet (ODS) after the test run that can be imported into Microsoft Excel, OpenOffice Calc, or any other spreadsheet program. The spreadsheet will include the raw test output and a series of graphs that show the performance characteristics of the drive over different I/O types, queue depths, block sizes, and read/write ratios. These graphs present the performance data in an intuitive format.

All tests are run using the FIO tool, and all executed test command lines and results are preserved in intermediate files for later examination if needed. As ezFIO is implemented as scripting languages, it is easy for users to examine and modify. All actual IO testing is performed by FIO, so there is no performance penalty often associated with scripting environments.

The ezfio performance test is run by executing a python script in the root directory of the git repo. To get the ezfio repo execute the following command from the user home directory:

>> git clone https://github.com/earlephilhower/ezfio

Interpreting the Test Results

Source: https://github.com/earlephilhower/ezfio

The content in this section is copied from the "ezFIO user Guide.pdf" Section 5 as a convenience to the reader. The ezFIO command save all the results to a spreadsheet.

Overview
The spreadsheet output will consist of three sheets: Graphs, Tests and Timeseries. The focus of this section is the Graphs sheet, as it provides a visual summary of the values recorded in the remaining sheets.

System and DUT Summary
The top of the sheet summarizes the system and device under test.

The next graph will show the results of the Sustained Mixed Read / Write test. This is a time-series graph and captures the IOPS delivered, second-by-second, by the device over a period of 20 minutes, as well as the numerical average and variance. Large variance or "choppiness" in the graph indicates that the applications are stalling while waiting for I/O.

Queue Depth
The next graph shows how the queue depth affects the Sustained Mixed Read / Write performance. Whereas the previous time-series graph provides a "best case" sustained performance by allowing for a high queue depth, the "IOPS vs. Queue Depth" graph will identify how an application with a lower number of I/Os in flight will perform.

4K Random Read Performance with Varying Queue Depths
The next two graphs shows the 4K Random Read performance over varying queue depths. The IOPS and total application I/O latency are charted. The latency graph often resembles a "hocky-stick", with the point of inflection occurring when the device reaches peak performance at the minimum queue depth. The increase of queue depth after this point often results in an exponential increase in I/O latencies, as the underlying device is fully utilized.

4K Pure Random Writes
The next two graphs are similar, but depict pure 4K Random Writes.

Random Read Performance / Sequential Reads / Identical Block Sizes
The next two graphs are useful for understanding how Random Reads perform compared to Sequential Reads of the identical block size. Usually, due to architectural implementation made by the NVMe device manufacturer, there are profound differences in Sequential (or streaming) versus fully Random performance.

Full Write Workloads with Varying Block Sizes
The next two graphs are similar but depict full Write workloads as the block size varies.

Software Configuration and Setup
To create a volume for ezfio performance testing the following command was executed on the command line:

>> sudo cio volume create -c 100 -o -t SSD --thick -v ezfioTest

This creates a virtual disk located in /dev/vdisk/vd# where "#" is equal to integer sequentially incremented based on the number of CSS 1000 drives install in the system. The virtual disk is mount to a specified directory. However, to perform the ezfio performance testing there can't be a directory mount to the virtual disk. To unmount the directory

>> sudo umount <full path to the mounted directory>

Now the virtual disk created using the cio software stack is ready for performance testing. To start the performance testing the following command is run in the terminal

>> ~/ezfio/ezfio.py -d /dev/vdisk/vd# -u 10 --yes -o ~/outputFIO

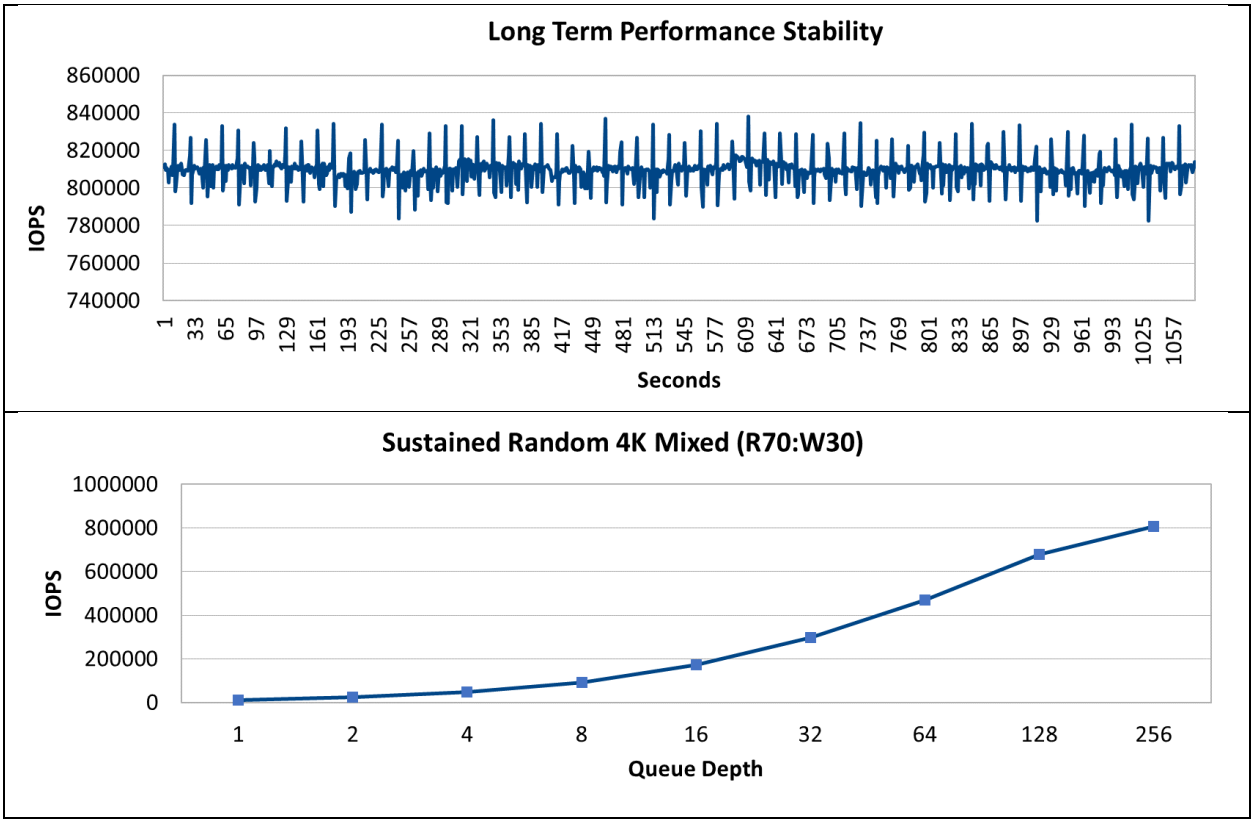
This command specifies the disk to be evaluated where -u specifies the percentage of the disk to be utilized and -o is the output directory for the results. The –yes is to acknowledge the warning of the ezfil tool.

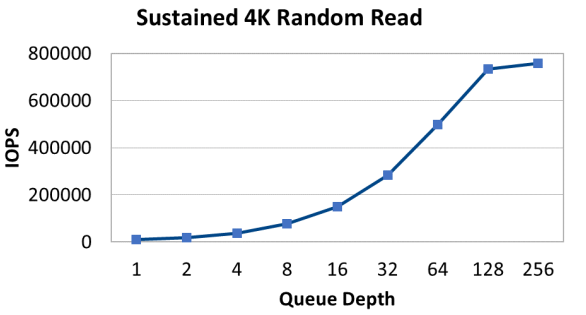
Results

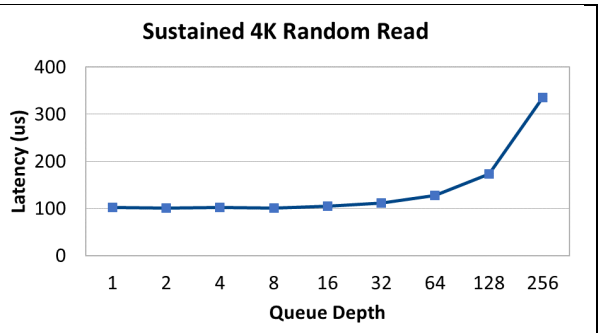
This section shows the individual performance graphs from the ezfio testing. A summary or aggregation of the results and graphs are shown in the Summary Section.

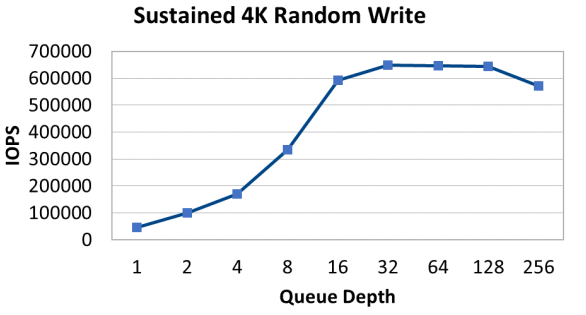
Single CSS 1000 ezFIO

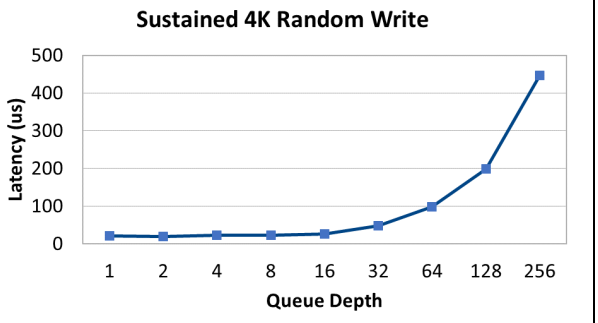
This section shows the results of the ezfio performance test running a single CSS 1000 directly (e.g., no cio software stack used for managing). The one of the four CSS 1000 cards were installed into the 20 PCIe Gen 3 x16 slot Klondike Platform.

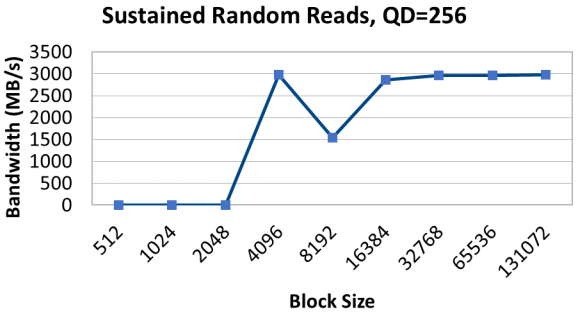


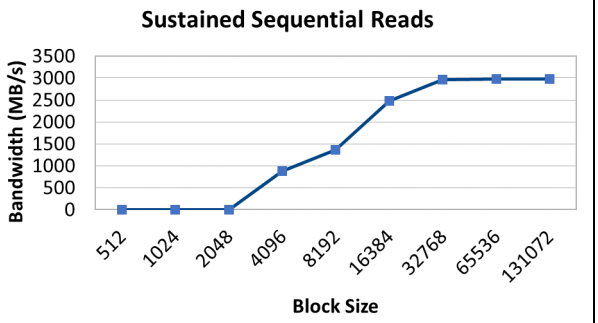


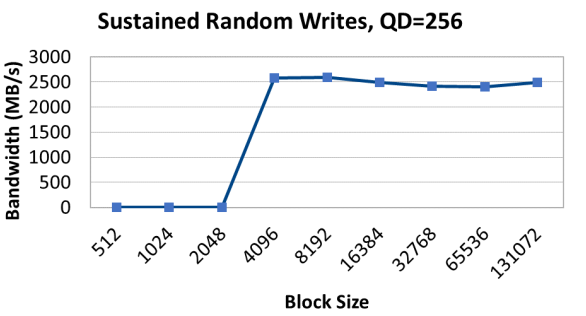


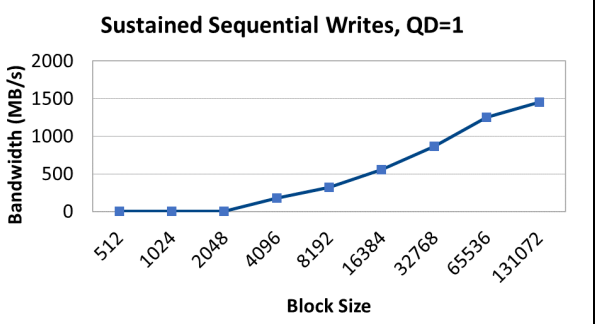


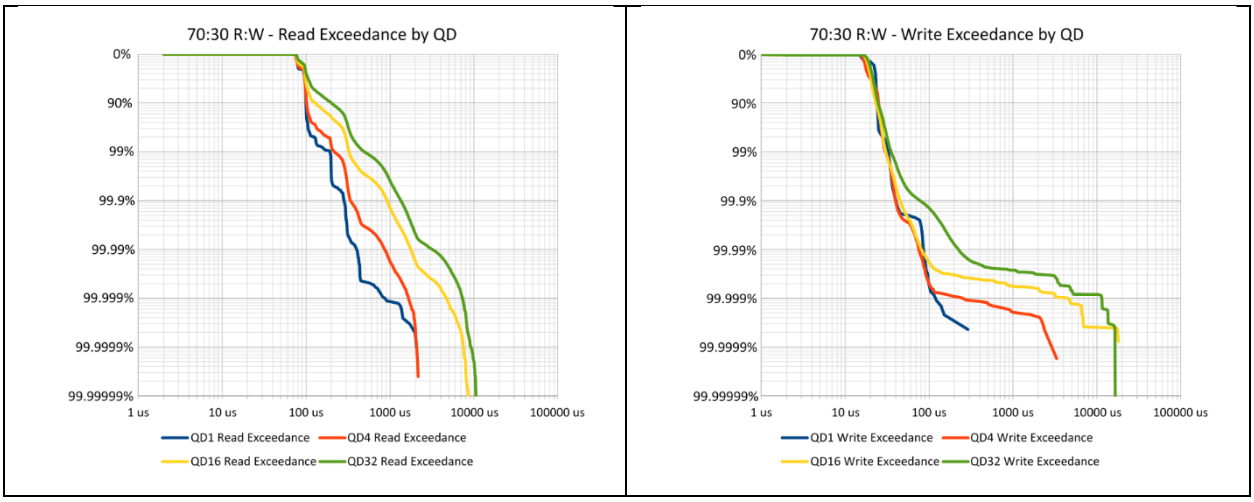












Clustered CSS 1000 (4x 6TB) Configuration 1

This section shows the results of the ezfio performance test running on a virtual volume created from the cio CSS 1000 cluster. The four CSS 1000 cards are installed into the 20 PCIe Gen 3 x16 slot Klondike Platform as seen in Figure 2

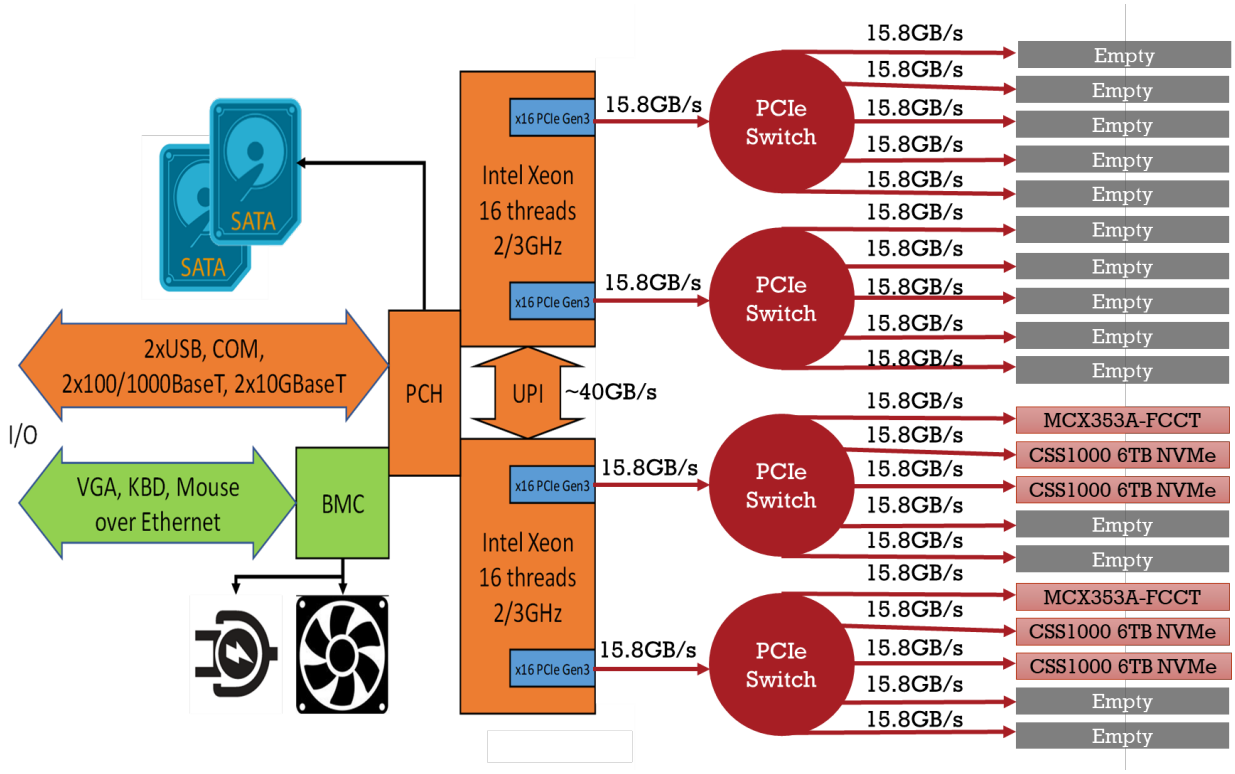
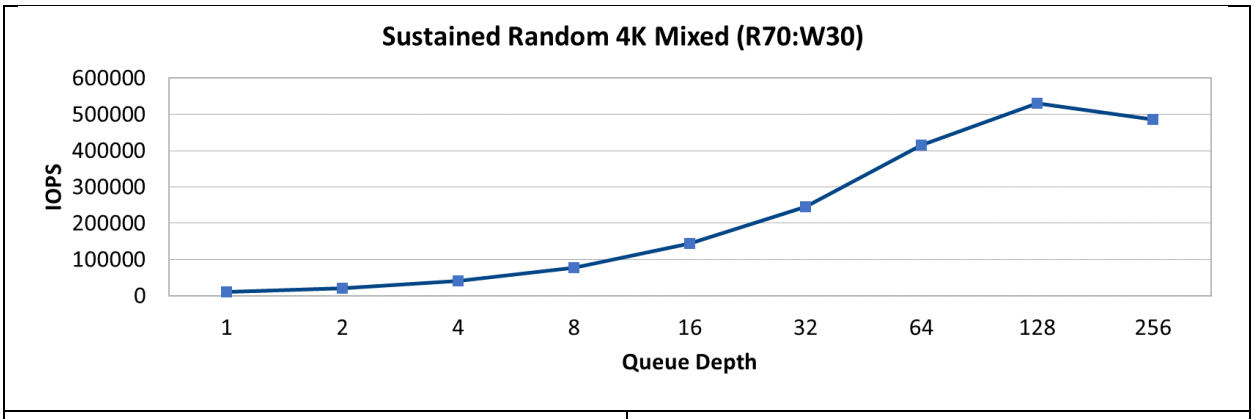
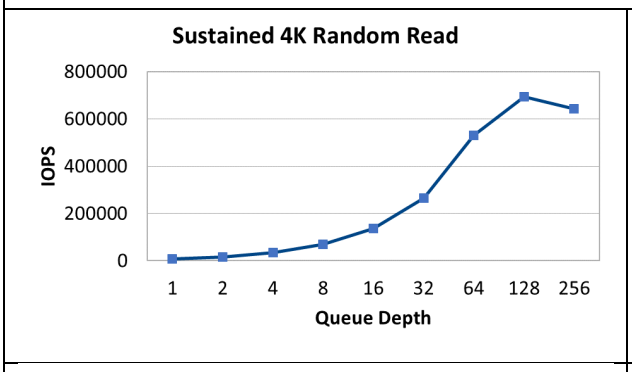
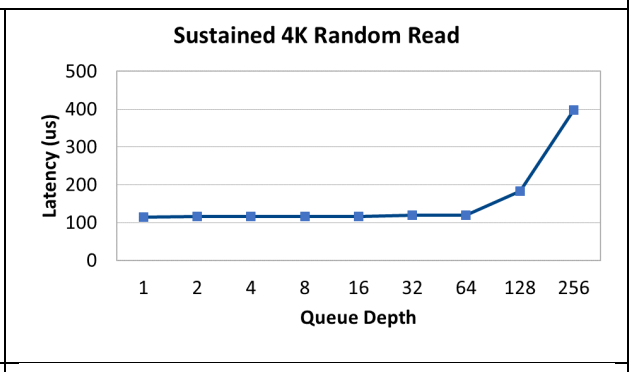
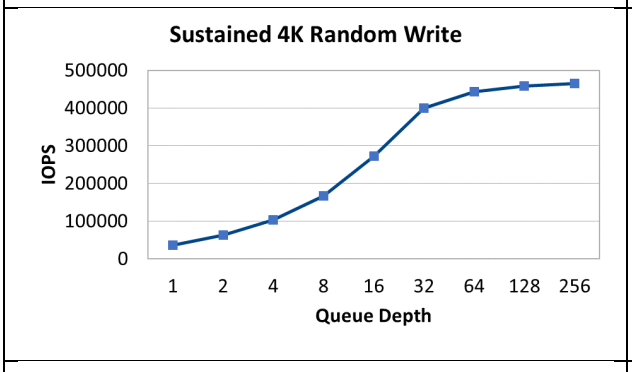


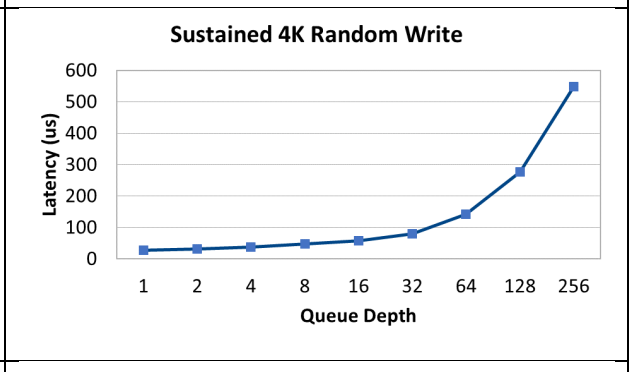
Figure 2: CSS 1000 configuration 1 in Klondike

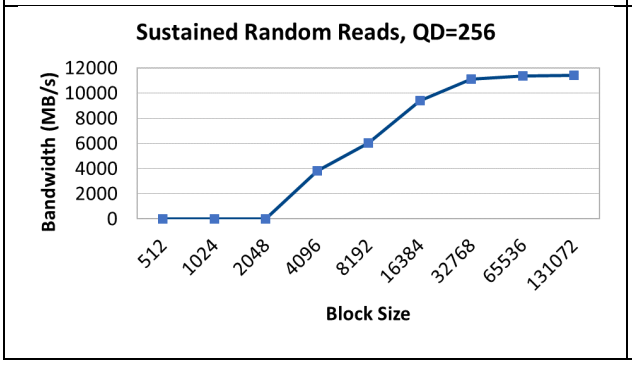


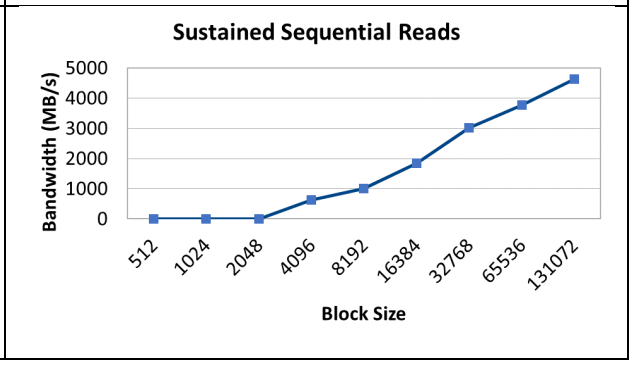


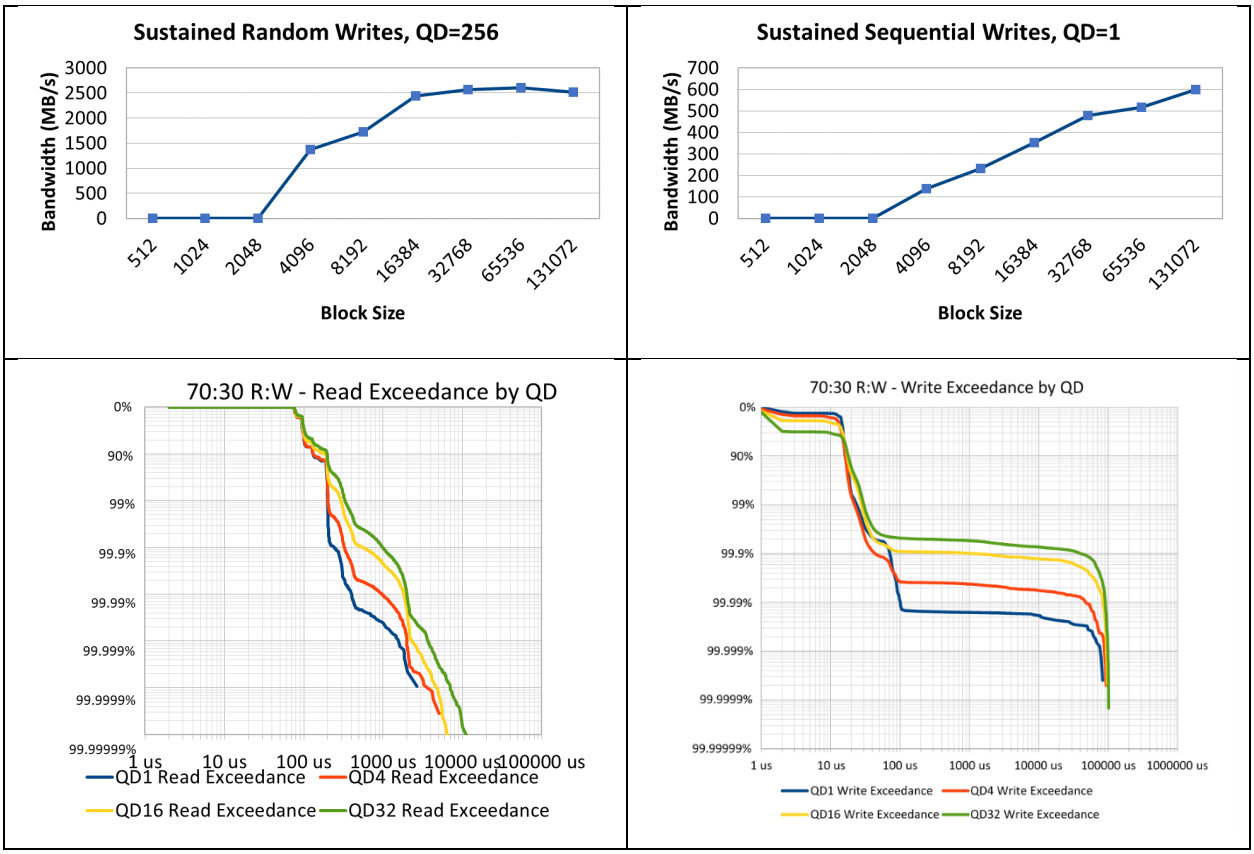












Clustered CSS 1000 (4x 6TB) Configuration 2

This section shows the results of the ezfio performance test running on a virtual volume created from the cio CSS 1000 cluster. The four CSS 1000 cards are install into the 20 PCIe Gen 3 x16 slot Klondike Platform as seen in Figure 3.

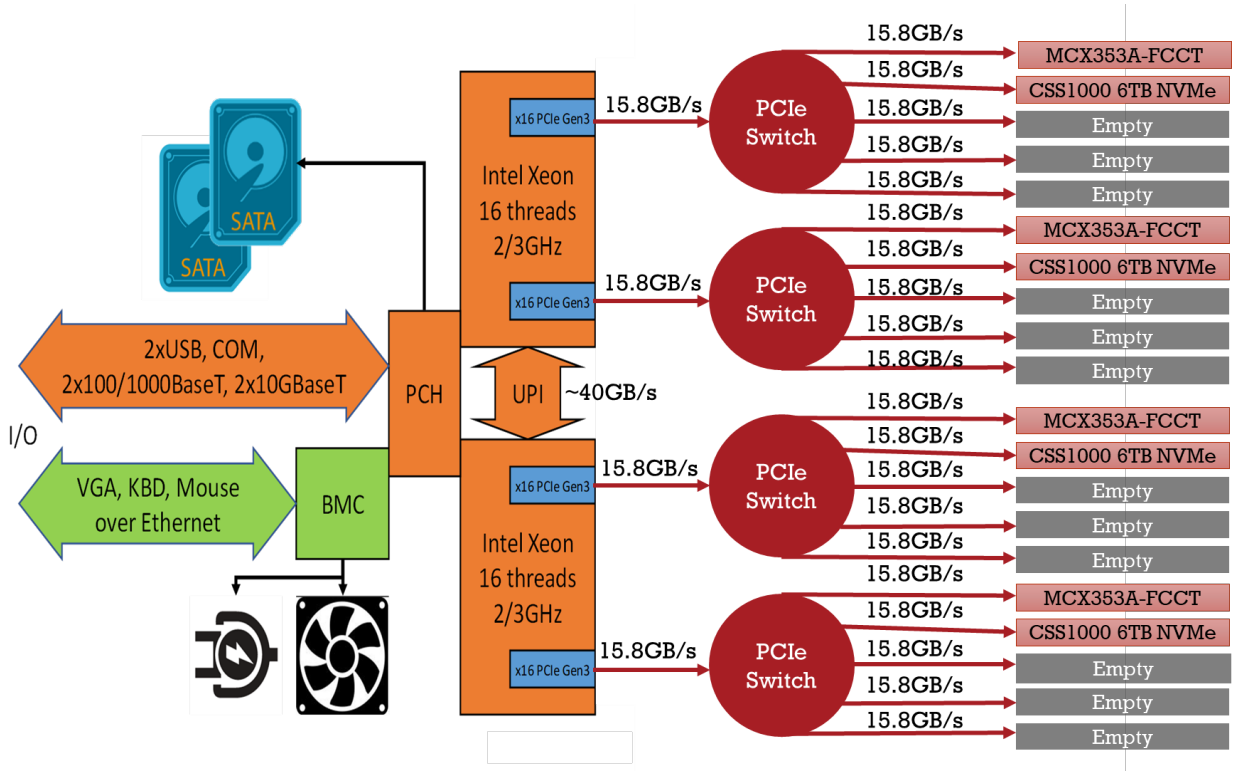
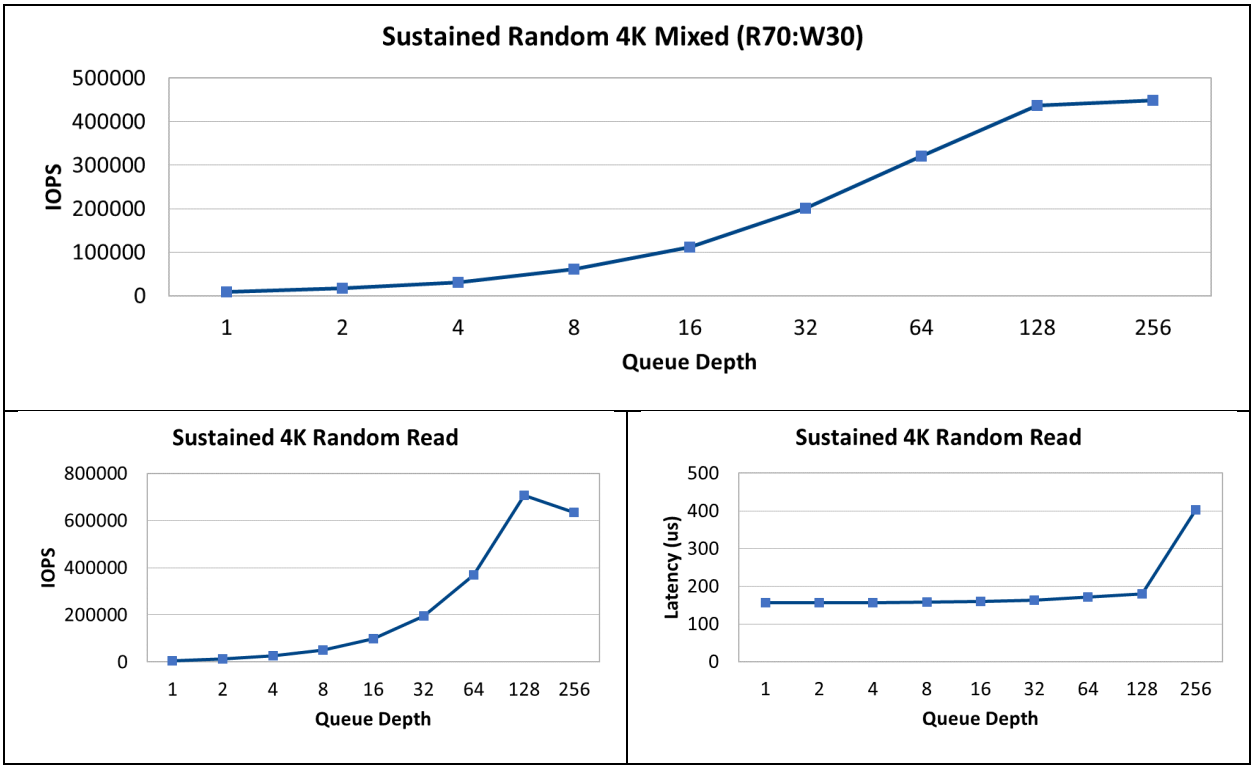
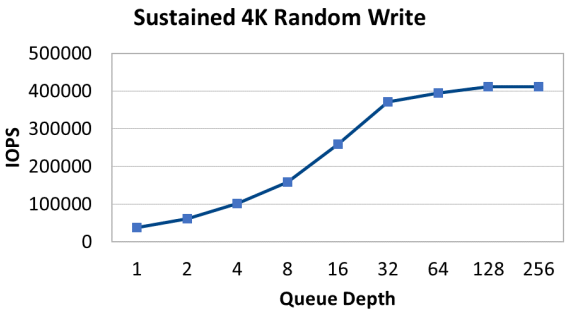
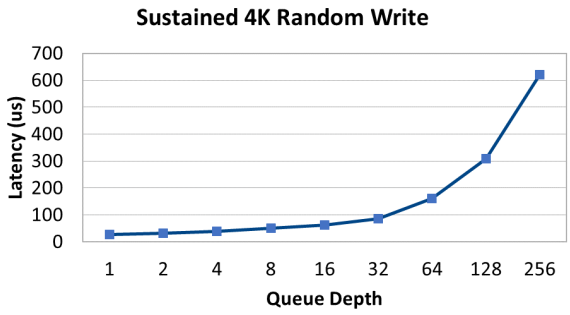
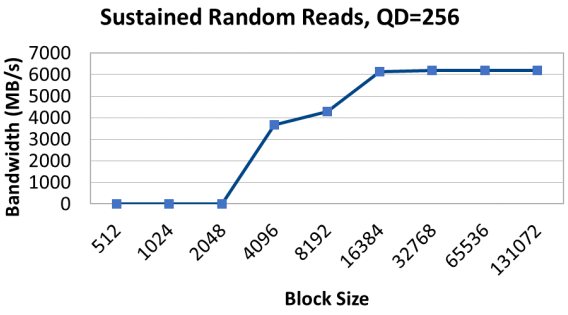


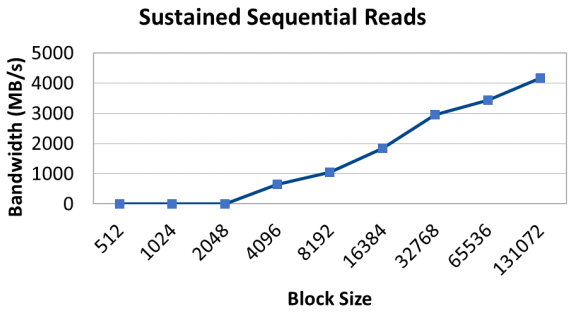
Figure 3: CSS 1000 configuration 2 in Klondike

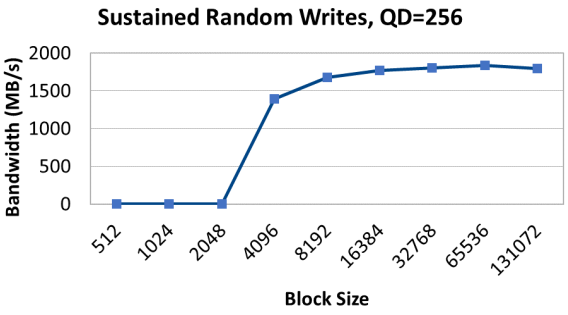


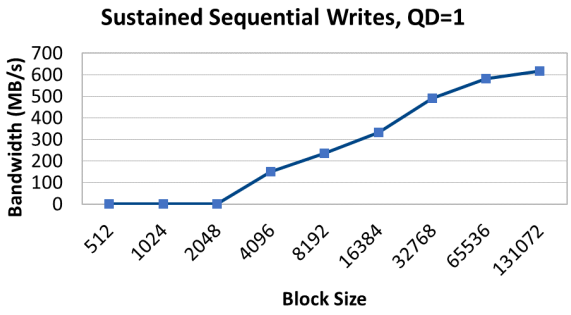


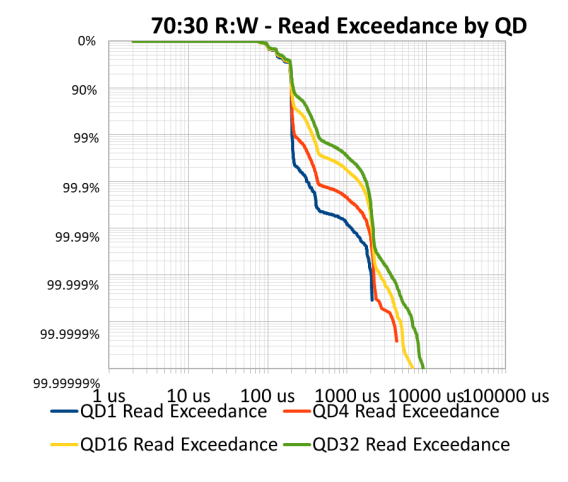


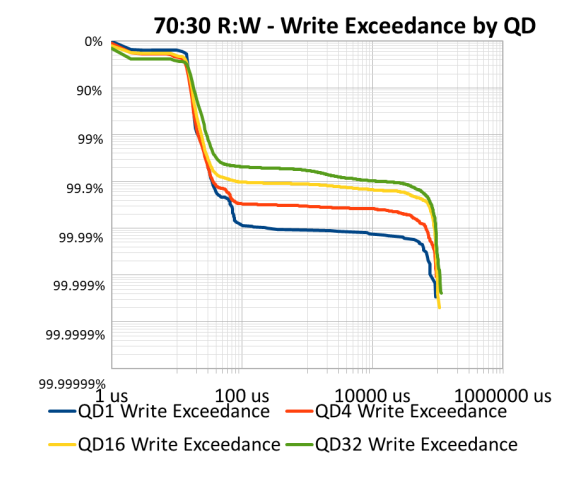












Clustered CSS 1000 (4x 6TB) Configuration 3

This section shows the results of the ezfio performance test running on a virtual volume created from the cio CSS 1000 cluster. The four CSS 1000 cards are install into the 20 PCIe Gen 3 x16 slot Klondike Platform as seen in.

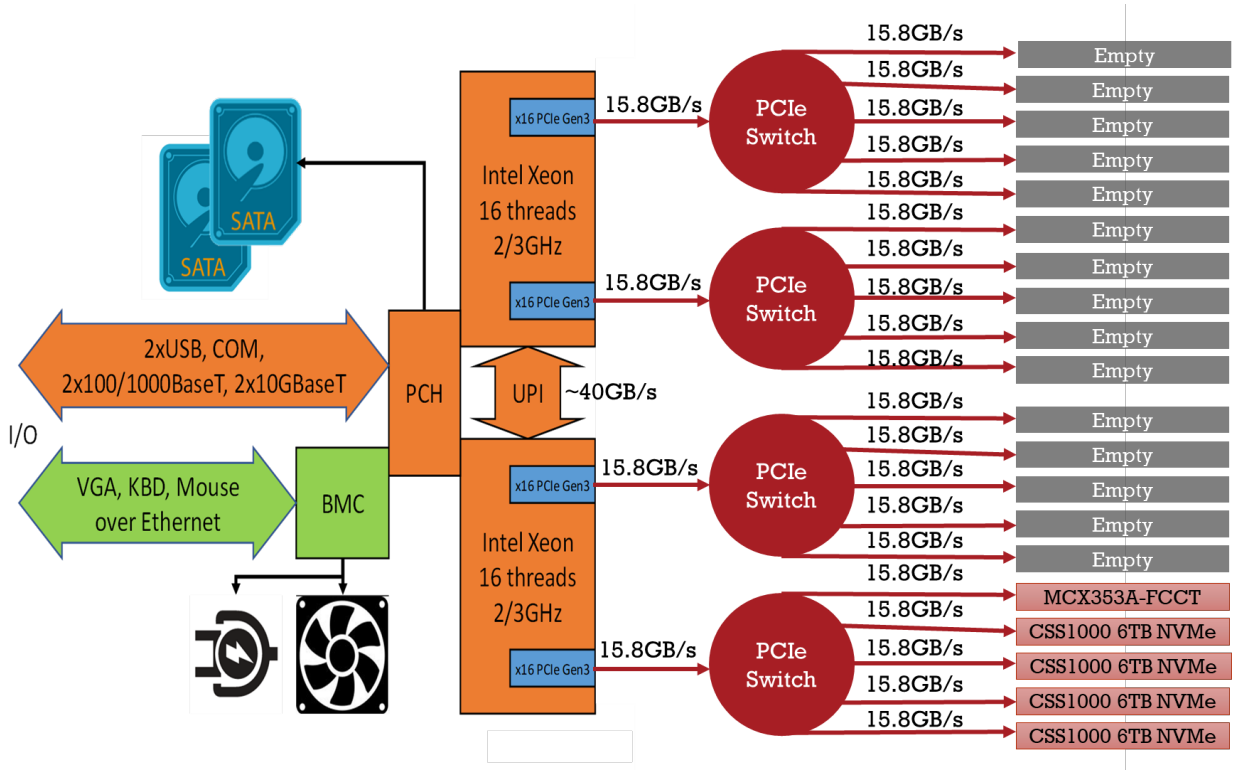
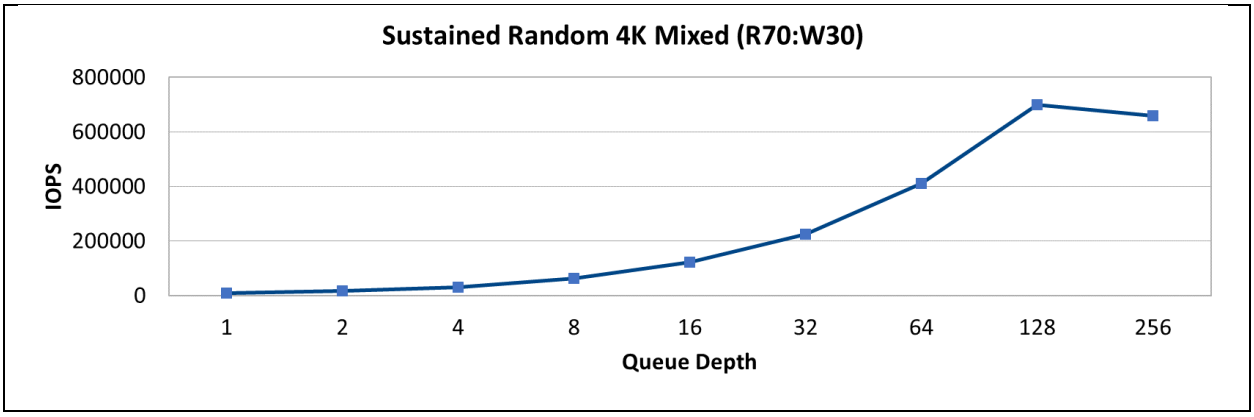
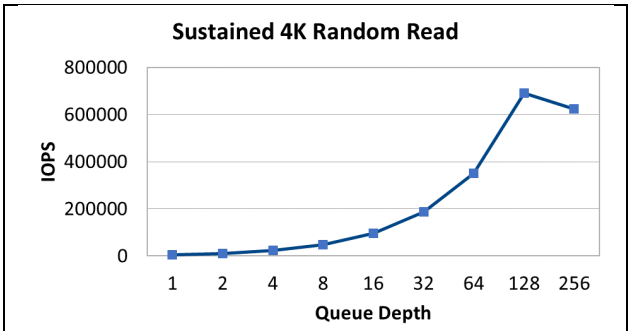
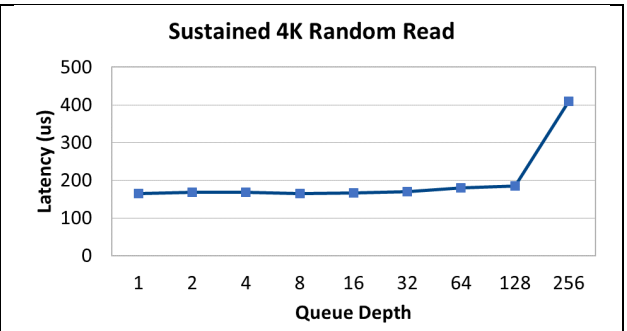
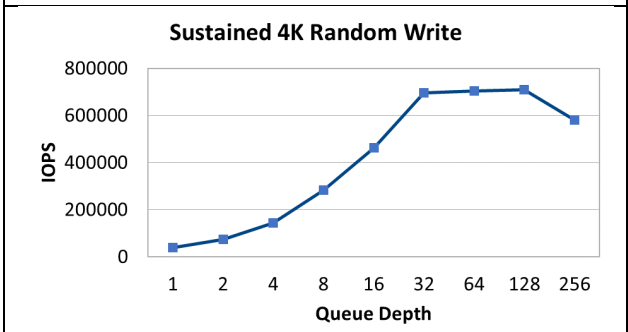


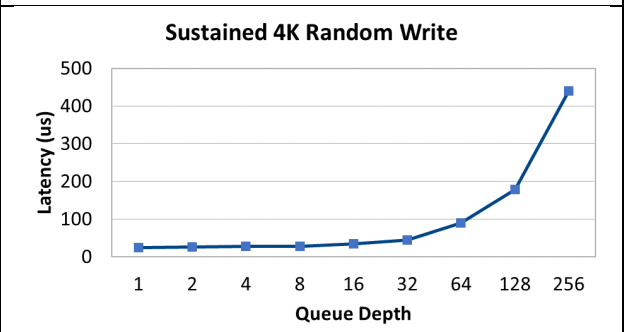
Figure 4: CSS 1000 configuration 3 in Klondike

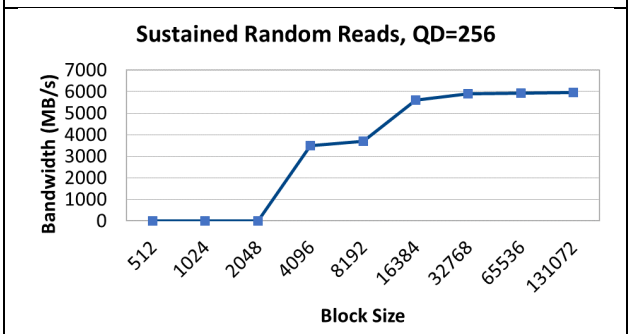


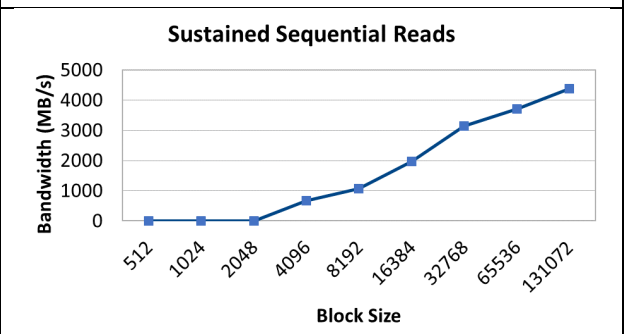


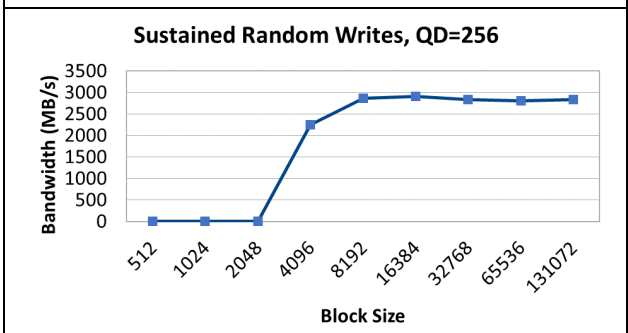


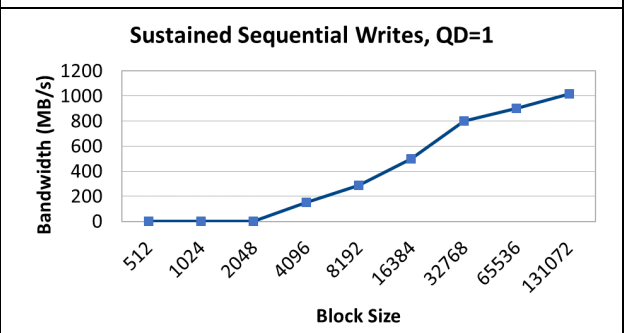


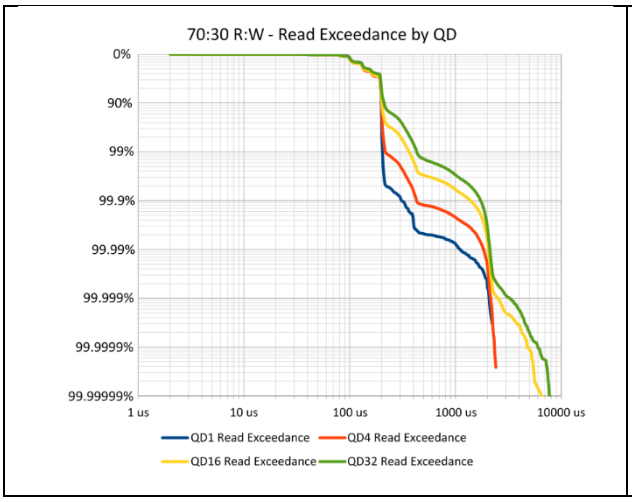


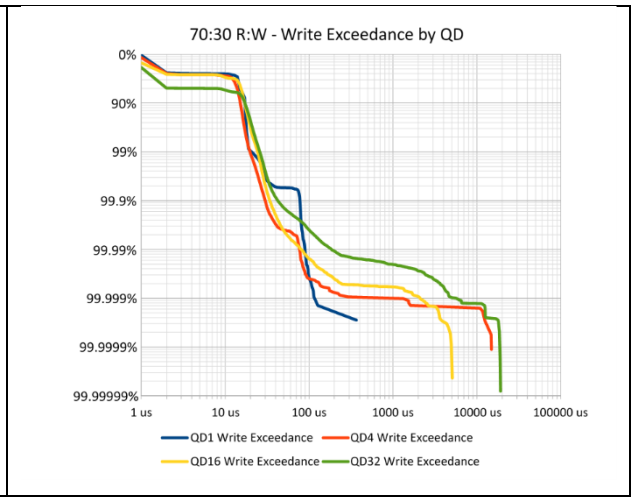












Summary

This section shows the aggregation of the 3 CSS 1000 configuration (using CIO) within Klondike along with a baseline performance of a single CSS 1000.