

The I/O Performance of the IceWEB Storage System

This paper focuses on the I/O performance capabilities of the IceWEB Storage System. It discusses the impact on I/O performance of various IceWEB features, and also demonstrates the effect of various configuration options. The information in this paper is useful for system sizing and configuration to ensure that applications get the storage performance they need to run at peak efficiency.

The primary benchmark used for this paper is the *IOMETER* utility. IOMETER is an I/O subsystem measurement and characterization tool (Figure 1). It is used as an industry-standard benchmark and troubleshooting tool and is easily configured to replicate the behavior of many popular applications. IOMETER runs on a client system and can read/write data to the storage array of a specific block size with either a sequential or random I/O pattern. The tool reports both I/Os per second (IOPS) and raw throughput (MB/sec). This paper primarily focuses on the total IOPS divided by the number of disk drives resulting in IOPS/drive. This is the key metric for system sizing, assuming that linear scaling is achieved as more drives are added to the storage pool. Linear scaling is not always achieved, and this paper also looks at other bottlenecks (such as bandwidth limitations) that constrain the linear performance scalability of the system.

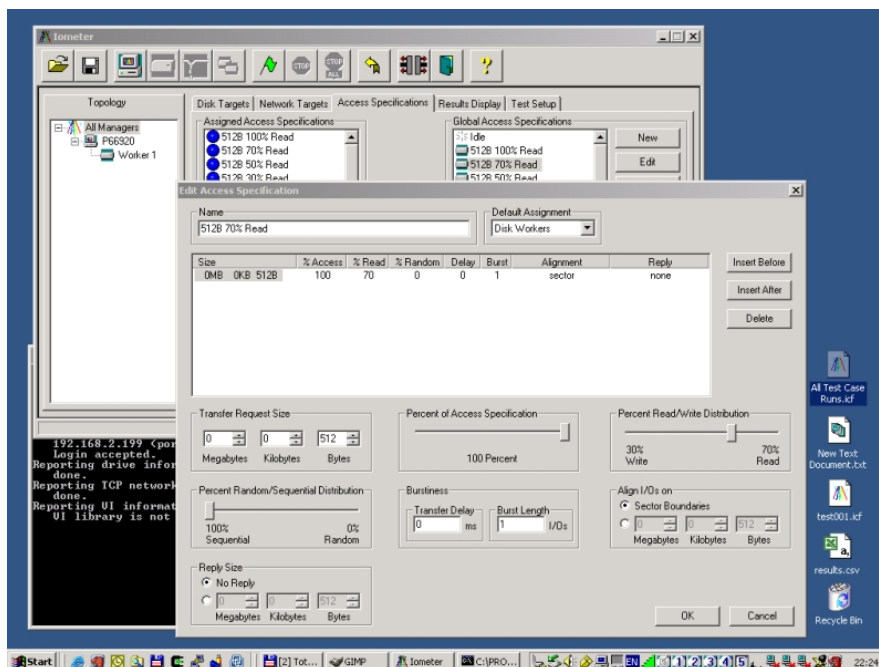


Figure 1 - IOMETER screen shot showing the definition of a work load

Test Configuration and Variables

Performance testing is always challenging in that there are many variables that affect performance and isolating those variables is key to understanding their impact. This paper focuses on seven main variables within the configuration:

- RAID settings
- # of RAID groups within a pool
- IOMETER block size
- Random vs. Sequential I/O
- ZFS block size
- Disk type
- Network (bandwidth) tuning

All the tests were run on a single IceWEB 3000HC with 24GB of RAM and twelve (12) 7200RPM 1TB SATA disk drives. IOMETER was run as a single client through a single 1GbE NIC on the host, through a switch, and then through a single 1GbE NIC on the array. All system defaults in terms of cache, the intent logs, network settings, etc. were used for the baseline tests (unless otherwise noted below).

IOPS Metrics & System Sizing

In most cases sizing a storage system consists of determining usable capacity and performance requirements of the applications using the system, and using this information to calculate the size of the array. Usable capacity calculations are straight forward and involve starting with raw drive capacity, accounting for various storage system overheads, and then determining how many drives are needed for the desired usable capacity. Performance sizing is more of a challenge. Performance is not a constant, so customers must decide whether to size for a peak or average work load. “Good” performance is also subjective; it relates to user response time for interactive applications and the execution duration for large application jobs. Some applications require outstanding performance (e.g. customer response time on a web site), whereas with other applications there is a lot of tolerance for lower performance (e.g. the monthly GL posting).

The simplest method for sizing the storage system for performance is to determine the I/Os per second (IOPS) required by all applications and calculate the bandwidth and the number of disks needed to satisfy that IOPS load. This method is appropriate for applications that perform small block random I/O to the storage system, and includes web applications, messaging, database applications, file shares, and most virtual server applications. For applications that utilize very large files or binary objects, throughput (MB/sec) is a more appropriate metric and will indicate how long various jobs may take. For the purposes of this paper, we focus primarily on IOPS-base sizing.

IceWEB has developed a sizing tool that takes IOPS and usable capacity as input and calculates the required number of drives. The tool also accommodates the typical overhead of RAID, deduplication, compression, and bandwidth. This tool is available via your IceWEB sales representative or channel partner.

IOPS vs. Throughput

The two metrics used for measuring storage performance are I/Os per second (IOPS) and throughput expressed as megabytes per second (MB/sec). Both these metrics measure the same thing; the amount of data transferred in one second. With IOPS, the key factor is the size of an “I/O”. I/O sizes vary considerably by application and file system, and range from “small” I/Os (e.g. 512 to 8192 bytes) to “large” I/Os (e.g. 64KB to over 1MB). Throughput can always be calculated from IOPS. For example, 500 1KB IOPS is equal to 500KB/sec or .5MB/sec throughput.

Disk drives have performance characteristics that favor large block sequential I/Os. The maximum throughput of a disk drive quoted by the manufacturer is determined by writing large blocks sequentially to the drive. Since the blocks are very large, the number of I/Os (blocks) per seconds are low, but the throughput (MB/sec) is high. Disks don’t do as well with smaller blocks, and so as the IOPS numbers increase, the throughput usually decreases. If you now start mixing together I/Os from different files and applications (random I/O), and mixed load of reads and writes, the drives cannot be as efficient and will slow down more. Thus the maximum IOPS values quoted by manufacturers are usually 512 byte I/Os read sequentially.

Most application environments primarily do small block random read/write I/Os. Even where applications handle large documents or objects, often the file system or the storage virtualization system will “randomize” the I/O. Therefore, as a rule of thumb, IceWEB sizes arrays using the following metrics for system sizing:

- 4KB random r/w
- 8KB random r/w

Unless otherwise noted, IOPS will refer to 8KB random I/Os with a 50/50 mix of reads and writes.

There are many published performance papers that discuss storage system sizing and quote rule of thumb numbers for drives. Figure 2 below shows typical non-vendor performance numbers that are often used for SAS and SATA drives. These are 8KB random I/Os and assume RAID and some degree of caching. These numbers also assume there are no network bottlenecks. As you can see, SAS is considerably faster due primarily to its higher rotational speed of 15K RPM compared to 7200 RPM for SATA.

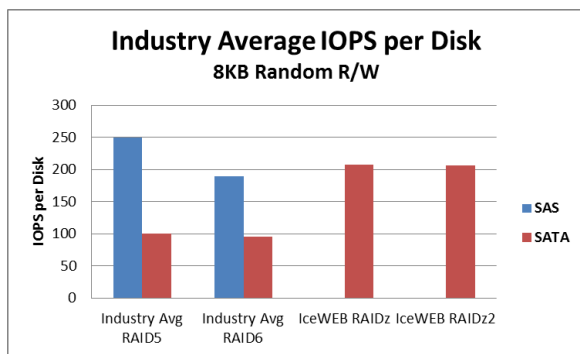


Figure 2 - Industry Averages for IOPS per Disk, SATA Drives

IceWEB baseline numbers for SATA are also shown for comparison (SAS numbers will be provided in a future version of this paper). You will notice that the IceWEB SATA numbers are about twice the industry average. The industry numbers have been in use for about 5 years, so the difference is mostly due to the more modern systems used by IceWEB (faster CPUs, multi-threading, more memory, better caching) and the specific write optimizations employed by ZFS.

RAID Settings

One of the major factors for performance is the underlying RAID settings and configurations used within an IceWEB storage pool. A storage pool is a collection of disk drives from which block volumes and file shares are created. Data from these objects are striped across all the drives in the pool, so not only does the pool deliver a specific capacity, it also delivers a certain number of IOPS that are shared by all the volumes and shares within the pool. Drives are added to the pool in RAID groups where the level of protection is selected. An obvious question is how big to make the RAID groups. The larger the RAID group, the less capacity it takes up for a given RAID setting... but, is there a performance tradeoff?

For example, a 12 drive pool could be implemented as one large RAID group with 11 drives worth of data and one parity drive (11+1). Or, the pool could be broken into smaller groups, for example, two 6 drive groups (2 x 5+1). Figure 3 compares the IOPS performance of these two configurations, and shows that the larger RAID group in fact performs about 10% better than two smaller RAID groups. The same result is seen with both single (RAIDz) and dual parity (RAIDz2) RAID groups. The conclusion is that for pools of around 12 drives, you should configure a single large RAID group if possible and this will maximize both usable capacity as well as I/O performance.

It is not necessarily the case that this “rule” scales up to any size pool. For example, it’s probably not true for large pools of 50 or 100 drives. Future testing will determine where the “peak” performance is for RAID group size.

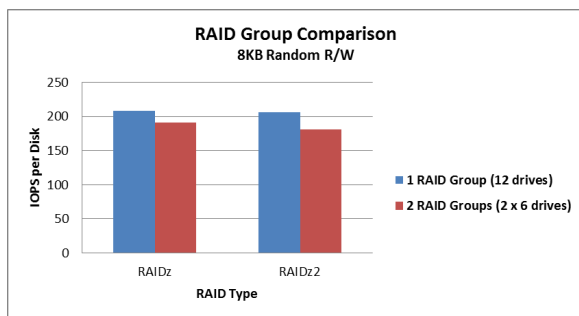


Figure 3 - RAID Group Configuration Comparison, SATA Drives

IceWEB OS (ZFS) Block Size

ZFS has the ability to specify the internal block size to use for a given volume or share. This block size is specified when the volume or share is created, and ranges in size from 1KB to 128KB. The best practice is to choose a block size that aligns with the dominant block size used by the application or file system

for I/Os to that volume. However, smaller block sizes within ZFS also mean that more memory is consumed for the virtualization tables, so a smaller block size may also impact performance if the system memory is insufficient for the amount of storage.

Figure 4 indeed shows that aligning the ZFS block size with the I/O size makes a difference for random I/O. For both 8KB and 16KB random I/Os against RAIDz disk groups, the best IOPS were achieved when the ZFS block sizes were the same as the I/O size.

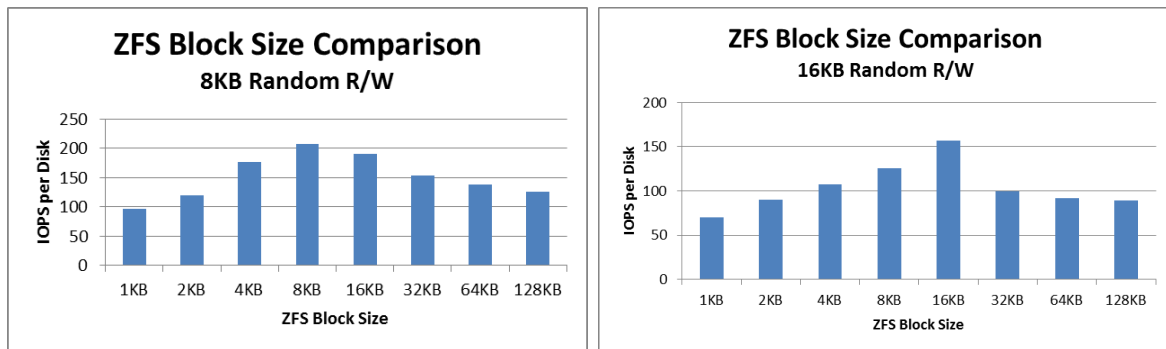


Figure 4 - ZFS Block Size Comparison, Random I/O, SATA Drives

Sequential writes should do best with a ZFS block size that is equal to or greater than what's being written. This is because the write I/Os should "pack" nicely into larger blocks and there will be slightly less overhead to larger block sizes. Figure 5 shows this to be the case.

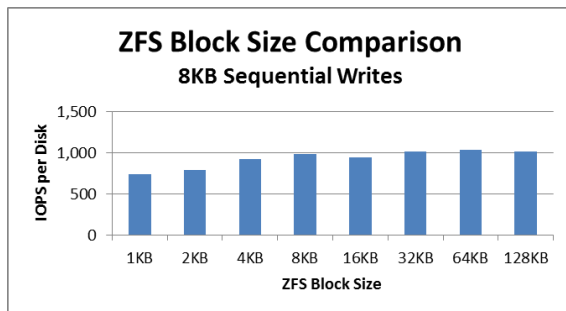


Figure 5 - ZFS Block Size Comparison, Sequential Writes, SATA Drives

In summary, it does help to align ZFS block sizes, especially for small-block random I/O. If you know the dominant I/O size for your application, you should use this setting for maximum performance. For example, both SQL Server and Exchange use 8KB blocks, and the I/O patterns are typically a random mix of reads and writes, so an 8KB ZFS block size is a good best practice for those volumes.

In cases where you do not know the I/O size your applications use, it's best to go with 8KB ZFS block size or larger. Larger block sizes (e.g. 64KB or 128KB) may be particularly important where the array has very large amounts of storage (>50TB), as this keeps the meta data tables smaller and enable them to stay cached in memory.

I/O Size

What is the performance of the IceWEB system given various I/O sizes? Figure 6 shows both the IOPS values and throughput for various I/O sizes. These charts show random R/W I/O using a “standard” storage configuration using RAIDz and an 8KB ZFS block size. If you know the average I/O size for each of your applications plus an estimate of how many IOPS (or throughput) is required, this data will allow you to determine number of drives needed for that application to deliver “optimal” I/O performance.

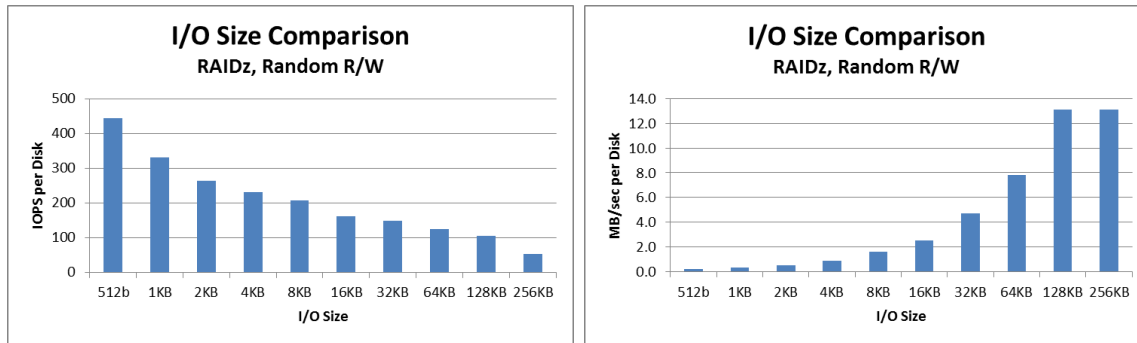


Figure 6 - I/O Size Comparison for RAIDz, Random I/O, SATA Drives

Notice that throughput is an inverse function of I/Os per second, as discussed in the first section of this paper. Also notice that the throughput flattens out between 128KB and 256KB block size. This is due to a network bottleneck being introduced ($12 \text{ drives} \times 12 \text{ MB/sec} = 144 \text{ MB/sec} = 1 \text{ Gb/sec}$) by the use of only one 1GbE NIC for these tests. The chapter on Network Optimizations below shows how to avoid this bottleneck.

For the IceWEB sizing tool, an environment requiring “3500 IOPS” is assumed by default to mean “3500 8KB random R/W IOPS using RAIDz” and a figure of roughly 200 IOPS per disk is used to calculate the size of the system. If the customer has more refined data concerning their IOPS load, the tool can account for this and produce an accurate estimate.

Random and Sequential I/O

If the application is known to be reading and writing sequential I/O streams, the data is quite different and is shown in Figure 7. The IceWEB storage system is optimized for writes, so the best case situation from an IOPS perspective are small sequential writes, and the data shows this with almost a 5x increase in IOPS for 8KB I/Os over a random mix of reads and writes. Notice the jump at 8KB I/Os over 4KB I/Os; this is due to the alignment with the 8KB ZFS blocks size, and from that point on the throughput stays relatively consistent due to the “I/O packing” discussed above.

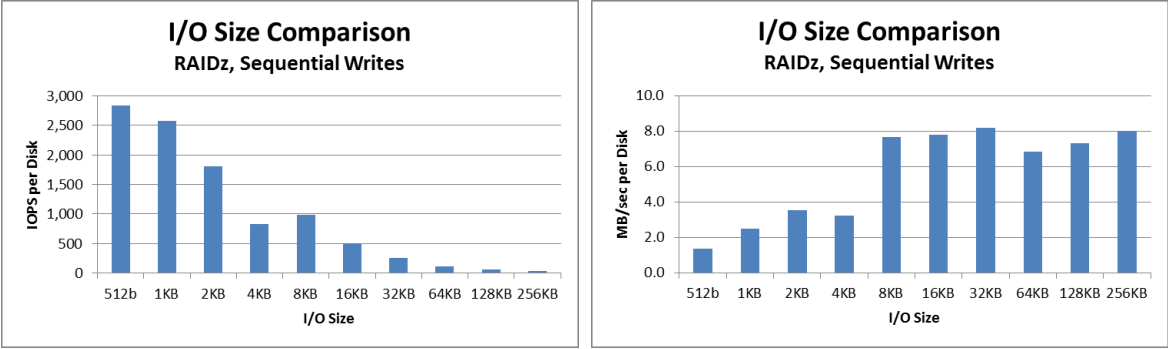


Figure 7 - I/O Size Comparison, Sequential I/O, SATA drives

Summary

The IceWEB Storage System 200 8KB random r/w IOPS for SATA drives; over twice the industry average of 95 IOPS for RAID5 storage systems. By deploying best practices in terms of RAID group configurations and networking, you can scale the system linearly with drives and obtain this IOPS value per drive to get to any desired performance level. The information contained in this paper is loaded into the IceWEB sizing tools and allows customers to accurately configure their arrays for the required application performance.