nexenta
Enterprise class storage for everyone

# NexentaStor

## NexentaStor ZFS Performance Guide

**Nexenta Solutions Engineering Team**

**September 2012**

**Table of Contents**

**Introduction** ..... **3**

**Goal of Document** ..... **3**

**Performance Characteristics of Nexenta** ..... **4**
Hard Disk Drives ..... **4**
Solid State Drive ..... **4**
Dynamic Random Access Memory ..... **4**

**Pool Configuration** ..... **5**
VDEVs ..... **5**
Data Redundancy and Protection ..... **5**
Logical VDEVs ..... **5**
RAIDZ2 ..... **5**
Two-way Mirror ..... **6**

**Pool Read Performance** ..... **6**
Why Mirrors for Read Performance? ..... **6**

**Hybrid Storage Pool** ..... **6**

**Working Set Size** ..... **7**
Performance Envelope ..... **7**

**Pool Writes** ..... **8**

**Hardware** ..... **8**

# NexentaStor ZFS Performance Guide

## Introduction

This document helps end users understand Nexenta and the sizing of storage appliances.

## Goal of Document

This paper walks through all the necessary knowledge needed for sizing storage nodes. Tuning an appliance should not be necessary if hardware is sized correctly for the workload. Once in a while, we might hit a few cases that require some tuning. However, 99% of the time, if the hardware is sized correctly, no tuning is necessary.

## Performance Characteristics of Nexenta

In order to describe the performance characteristics of a storage appliance, we will characterize the performance for individual components and then walk through how they all come together to maximize performance.

### Hard Disk Drives

Individual hard disk drives (HDD) are limiting due to the number of IOPS (Input/Output operations per second) delivered. IOPS from an HDD is determined by the rotational speed of the platter. HDDs come in a variety of speeds ranging from 5400 – 15000 RPM. Due to its higher RPMs, a 15000 RPM drive can deliver nearly 3x the IOPS that a 5400RPM HDD can deliver.

In simplistic terms, because an HDD is a spinning platter, the platter is limited to the number of times it can rotate and return to the same vector where it started. If an HDD spins at 7200RPM, it will return to the same place 120 times as shown by the formula below. This gives a better idea why random IO performance is limited.

7200RPM / 60 seconds = 120 IOPS

Like most storage arrays, NexentaStor has the ability to combine multiple HDDs to increase the aggregate number of IO operations. But due to physical limits of cable runs and cabinet size, realistically you are limited to 480 drives per system—that equates to roughly 60,000 random read IOPS from the base pool using mirrors.

In most storage applications, we see 7200RPM and 10000RPM drives giving a good balance between cost and performance.

### Solid State Drive

Unlike HDDs, there are no moving parts to a solid state drive (SSD). Latency, or the time it takes to complete an IO operation, is extremely low for SSDs. This allows an SSD to deliver more IO operations in a shorter period of time than an HDD. For an enterprise grade SSD, this can be upwards of 100,000 random IOPS at 4KB. One SSD can deliver the IOPS of 1000 HDDs—a massive improvement in IOPS delivery.

### Dynamic Random Access Memory

Dynamic Random Access Memory, also known as DRAM, is one of the fastest available storage areas on the system, with the ability to deliver one million IOPS. Unfortunately, the data stored in DRAM is volatile but, nonetheless, extremely fast and very useful.

## Pool Configuration

The configuration of the storage pool determines the performance of the appliance. Although drive count aids in increasing performance on the storage appliance, it does not determine the performance characteristic.

### VDEVs

Simply put, VDEVs are virtual devices—they can be made up of many things, such as RAM disks, disk slices, plain files, etc.

In the context of a Nexenta storage appliance, VDEVs are virtual devices made up of whole disks. Only **whole** disks are allowed to make up a VDEV.

Pools are created by striping data across logical VDEVs. Logical VDEVs contain multiple disks arranged in different data redundancy schemes.

### Data Redundancy and Protection

Top-level VDEVs, or root VDEVs, are made with tree structures: the root being the pool and the logical VDEVs contain the physical disks that make up the leaf nodes. The logical VDEV is arranged to allow for redundancy using parity or mirrors, also known as RAIDZ and mirrors. The most common configuration seen in logical VDEVs are RAIDZ-2 and 2-way mirror configurations.

### Logical VDEVs

A root VDEV is made up of logical VDEVs. The number of logical VDEVs determines the IO performance of a pool. IOs written to the pool are written to different VDEVs roughly every 1MB. Logical VDEVs are seen as one unit.

### RAIDZ2

RAIDZ-2 VDEVs are made up of data disks and two parity disks. Ideally, we would like to see an even number of data disks, with the most common number being six or eight total disks in the VDEV, allowing for four or six data disks and two parity disks. Anything bigger increases the rebuild time for the VDEV in case of a failed disk.

In a RAIDZ-2 configuration, a single IO coming into the VDEV needs to be broken up and written across all the data disks. It then has to have the parity calculated and written to disk before the IO could complete. If all the disks have the same latency, all the operations to the disks will complete at the same time, thereby completing the IO to the VDEV at the speed of one disk. If there is a slow disk with high latency in the RAIDZ-2 VDEV, the IO to the VDEV does not complete until the IO on the slowest drive completes.

### Two-way Mirror

Mirrors have roughly the same performance as RAIDZ-2 on writes. In mirrored VDEVs, data has to be written to both disks before the IO is complete. If one disk is slower than the other, the write does not complete until both copies are written. Therefore, write speed is the speed of the slowest disk. On reads, both disks in a mirror can be used for separate IO—giving double the speed of a single disk.

## Pool Read Performance

Pool performance is determined by the number of logical VDEVs. Each logical VDEV has the IOPS of the slowest disk in the VDEV. Therefore, the IOPS of the pool for random reads is calculated as follows:

Read on RAIDZ-2:
Pool Read IOPS = Logical VDEVS * Slowest Member Disk IOPs

Read on two-way mirror:
Pool Read IOPS = Logical VDEVS * Member Disk IOPS * 2

### Why Mirrors for Read Performance?

The number of disks a JBOD holds is predetermined. A 24-bay JBOD can hold only 24 disks.

A logical VDEV made up of mirrors can be made up of two disks. A 24-disk JBOD could have 12 logical VDEVs if VDEVs are made of mirrors, or four logical VDEVs if they are made of six disks in a RAIDZ-2. Twelve VDEVs versus four goes back to the formula that determines pool performance. The more logical VDEVs we have in place, the more we generate IOPS. In addition to having more logical VDEVs with mirrors, we have the added benefit that reads from mirrors will give us twice the performance of RAIDZ-2 in terms of IOPS.

## Hybrid Storage Pool

Putting it all together, we can combine pool speed with SSD and DRAM for caching to increase the performance of the appliance.

The key to NexentaStor optimization is the Hybrid Storage Pool (HSP). DRAM is used as an Adaptive Replacement Cache (ARC), efficiently storing both frequently used and recently used data. The Separate Intent Log is used to optimize latency of synchronous write workloads, such as NFS. The log satisfies synchronous write semantics while the transactional object store optimizes and allocates space on the main pool storage.

The log does not need to be very large in size and is structured according to the amount of data expected to be written in five seconds, the current time allocated by default to commit transaction groups. A Level-2 ARC (L2ARC), or cache device, can be used to cost-effectively grow the size of the ARC. For large, read-intensive workloads, the cost-per-gigabyte of SSDs is lower than main memory DRAM. Excellent read system performance can be achieved using modestly priced SSDs. The main pool performance can be less critical when the system is configured with enough RAM, and the log and L2ARC devices are fast.

## Working Set Size

The Working Set Size (WSS) is used to describe the amount of space most commonly used for applications or workloads. The use of WSS to describe performance is becoming increasingly useful as disk sizes increase. In many cases, systems are configured with 10s or 100s of terabytes of storage, while only a small fraction of the space is used at any given time. This fraction is the working set size.

### Performance Envelope

A performance envelope is used to describe the maximum limits of system performance. This concept is familiar to car or airplane enthusiasts: maximum horsepower, acceleration, altitude ceiling, etc.

For NexentaStor, the boundaries of read performance are the speed of RAM, the speed of the L2ARC, and the speed of the pool devices as mentioned earlier. The size of RAM and L2ARC caches can be adjusted as needs warrant—an obvious opportunity for cost/performance optimization.

Each of these variables can be measured in the lab or, in the case where the system is being designed on paper, gleaned from component data sheets. Each component can be measured easily under laboratory conditions with off-the-shelf benchmarking tools.

The expected maximum performance is based on the probability that a read IO operation is accessing data that already is cached in faster memory or L2ARC.

Mathematically, we can say that the expected maximum performance approaches the performance of the pool disks for an infinite working set size.

If the entire working set can be cached in RAM, then the performance is the best. If not, then L2ARC devices can be used to effectively increase the size of the cache.

Notice that RAM is much faster than the L2ARC devices. It is preferred that DRAM be fully populated before adding L2ARC devices. Also, L2ARC devices are approximately an order of magnitude faster than pool devices.

## Pool Writes

ZFS provides a Copy on Write (COW) technology, and each write is associated with a transaction group (TXG). TXGs have three states: open, quiescent, and writing. Writes go to RAM and are written to disk sequentially. TXGs sync to disk every five seconds or when 1/8$^{th}$ of system memory is filled. Write performance depends on the bandwidth of a VDEV. As mentioned above, TXGs are synched to disk at approximately 1MB per VDEV.

All writes are allowed at full speed to memory when a TXG opens. If the pool cannot keep up with the amount of writes from a single TXG sync, we then see a condition called write throttling. Keep in mind that on a busy system, we may see hundreds of megabytes of data being written in a single transaction. Write throttling was put into place so a system can provide reads and writes without IO in a wait state. Currently, if IO hits write throttle, each write system call is delayed by one tick.

## Hardware

When selecting hardware, it is important to choose items that are on the *Nexenta Hardware Compatibility List* (HCL). Storage nodes require balance when designed. There is a relationship to the amount of ARC, L2ARC, write speed, read speed, and capacity. To be more economical, understanding data and working set size improves IOPS vs. capacity. In many systems, we see storage systems running out of IOPS before capacity. It may be wiser to consider using smaller capacity drives, and more of them, to give better IOPS vs. capacity ratios. Nexenta publishes standard configurations for partners and resellers that give great balance in terms of performance and capacity.