Global Leader in Software Defined Storage

Nexenta Technical Sales Professional (NTSP)

COURSE CONTENT

Nexenta Technical Sales Professional (NTSP) Course

USE CASE:
SQL DATABASE

Use Case

SQL Database

Objectives:

- General characteristics
- System Design
- Disk Layout
- Optimization

General Characteristics

- I/O and latency are the key
  - mirrors or tripple mirrors are the best
- Typical small block sizes (bad for RAID-Z)
- Very often high amount of random read I/O
- Writes are mostly sync writes or async writes followed by fsync
- The more spindles the better I/O performance
- Often not much space required
- Sometimes more disks are needed to reach the required I/O then required from space usage
- Throughput is unimportant
- 10k/15k disks or SSD's as data disks will help on high random read workloads and small data footprints

System Design

- CPU
  - The faster the better
  - Prefer faster core speed versus number of cores
- Memory
  - Minimum 128GB
  - Larger memory for Read intensive environments
- Network
  - 2 x 10Gbe – Determine if SFP+ or Copper
- HBA
  - One JBOD per HBA if possible

Disk Layout

- Pools
  - 1 or 2 Pools
- Redundancy Type
  - Mirror
    - 1+1 (typical 300GB to 1TB)
  - Tripple Mirror
    - 1+2 (typical 600GB to 3TB)
  - The more vDevs the better
- ZIL/SLOG
  - Mirrored Pairs
    - Low Latency e.g. ZeusRAM or good SLC SSD's because of sync write amount
    - If two pools make sure you have 2 mirrored pairs

      Why mirrored? Because if you loose one SLOG the performance will drop dramatically on sync writes which could make database nearly unresponsive.
- L2ARC
  - 1-4 200 or 400GB SSD's smaller and more is better
  - Make sure you have enough memory to handle larger L2ARC builds
  - The more DRAM the lower L2ARC can be by same working set size
  - Allways add more DRAM to the server first, 512GB could be good solution

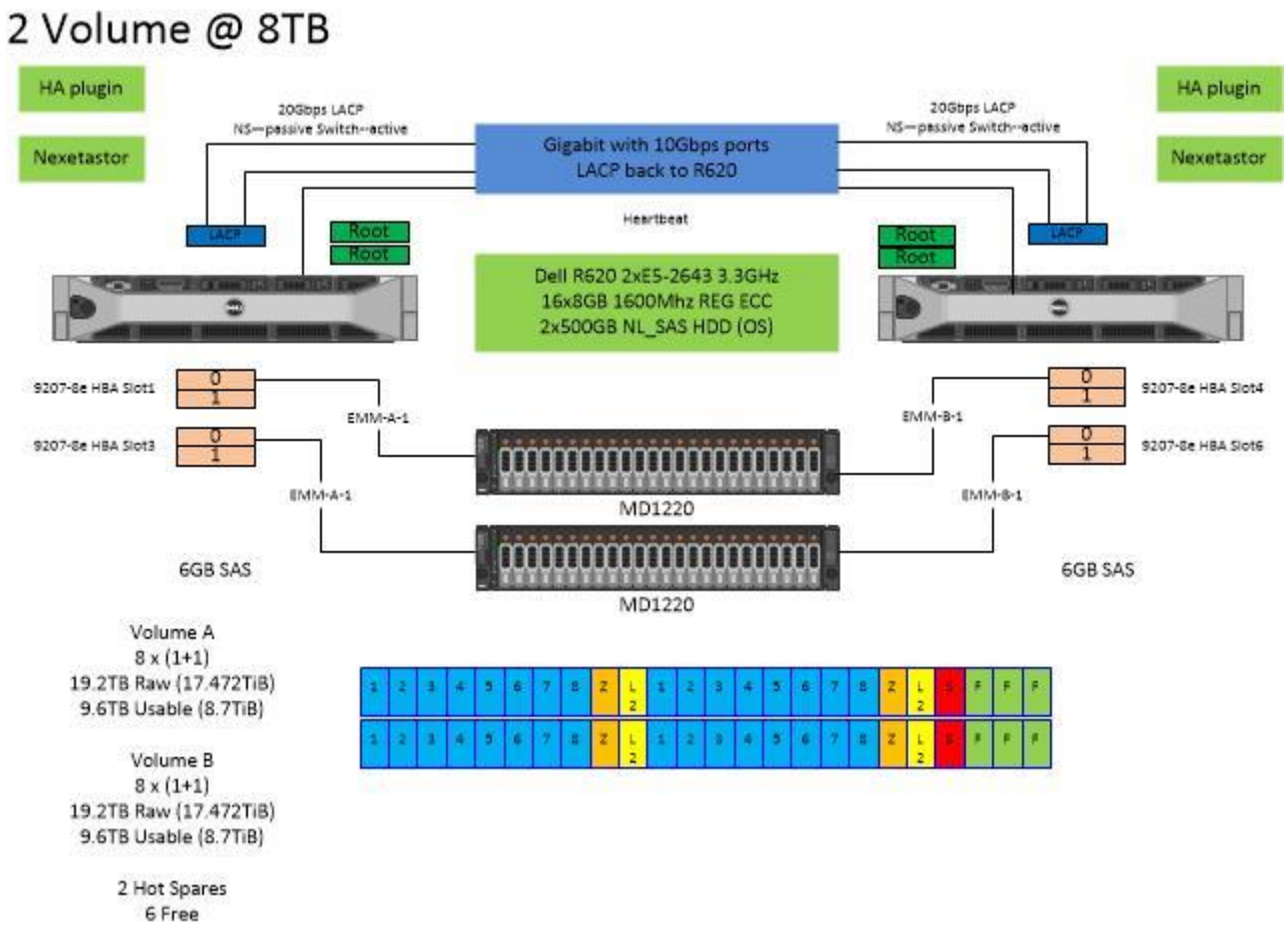
Optimization

- Record Size
  - Match with database is the best
  - More shares or zvols will increase performance at least 4 zvols is a good start
  - NFS with several shares or iSCSI with different zvols and each onesr ecord/blocksizes fitting to database blocksise are the best
  - Exampe: Oracle Database 32k for data and 8k for redo log
  - Example: Mysql/InnoDB 16K for data and 128k for log
- 2 Pools, one on each NexentaStor Server will utilize both nodes DRAM for caching. We can have 1TB of DRAM in a cluster
- Propably adding more memory to the database server could be more efficient then adding it to the NexentaStor Servers
- Keep Pools allways below 80% fill rate
- By default activate LZ4 compression but disable it if database is allready compressing the data blocks
- The more disks the better the I/O with mirror or tripple mirror configs
  - Assume for random reads on data disks
    - 7.2K = 80 I/O
    - 10K = 125 I/O
    - 15K = 180 I/O
    - SSD = 20000 I/O
- Don’t forget the throughput completely even it is less important

  Example:

  10000 I/0 with blocksize 32k are 320MB/Sec so this is never possible over 1G Ethernet

SQL Database Example with 1.2TB 10K disks



Module Quiz Questions

- SQL Databases are always high throughput environments?
- What is the preferred Record Size?
- What Redundancy Type is recommended?
- I need more Read Cache should I add more DRAM or more L2ARC first?
- Is a mirrored pair of SLOG increasing the write performance compared to a single SLOG? (while normal operation without faults)