Part 3 Storage Performance and IO Load Basics

Chuck Paridon
Member SNIA GSI
The I/O Chain

Queues of requests

Controller bottleneck

Adapter bottleneck

Load imbalance

Host

Adapter

I/O Bus

Controller

150 MB/Sec

80 MB/Sec

100 MB/Sec

70 MB/Sec

8-20 MB/Sec
Important Workload Considerations

Workload Parameters

- Block Size
  - The block size is the quantum if data requested for each IO transaction
- Read/Write Ratio
  - In general write IOs are “more expensive” than reads
  - Because these operations treat resources differently, the ratio matters
- Access Patterns
  - In practice, access patterns are as numerous as the stars. We consider:
    - Random and Sequential
- Working Set Size
  - An Application’s Working Set is the total address range traversed during it’s operation
  - The importance is amplified if there is sufficient cache to contain a considerable portion of the set size
- IO Demand Rate
  - For small block transfers (<64 kB) this is IO’s/sec (IOPS)
  - For larger block transfers this is in MB/s
Important Workload Considerations

- **Workload Parameters**

  - **Queue Depth or Demand Intensity**

    - This parameter represents the degree of IO demand parallelism presented to the storage by the application.
    - The influence of this item is quite profound, especially when using a random, small block workload.
    - Unless there is a sufficient demand on the LUNs, the addition of more spindles has no effect on performance.
    - This is another “difficult to estimate” parameter.
    - The estimate for this parameter is best gleaned from the application vendor or the system administrator.
    - Some applications, such as Oracle, allows the DBA to select the degree of parallelism employed during table scans, etc. So, the DBA is best source of this data.
Memory Performance Hierarchy

- **Memory (host)**
  - Very fast, small, expensive
  - Good for frequently accessed data read and write
  - Closest to CPU, least access time
  - Under control of the application or OS “lazy” IO issue
  - IO aggregation (write coalescing, read ahead)

- **Cache (controller)**
  - Very fast, small, expensive
  - Good for frequently accessed data
  - Read: helps some workloads, depends on application
    - Not terribly important for random access workloads
    - Cache efficiency very important
    - Prefetching and sequential write coalescing
  - Write-back: helps with most workloads

- **Disks**
  - Slow, but very large and inexpensive
I/O Queues

**One Queue per Device (LUN)**

Queue depth = 2
- a8
- a7
- a6
- a5
- a4
- a3
- a2
- a1

Queue depth = 1
- b8
- b7
- b6
- b5
- b4
- b3
- b2
- b1

Queue depth = 4
- c8
- c7
- c6
- c5
- c4
- c3
- c2
- c1

**Host**

- Queued
- Posted
- Completed

**Array**

- Queued for processing
- Currently processed
- Posted to backend
- Completed (cache hit)

- Posted to backend
- Completed (cache hit)
- Posted to backend
- Posted to backend
Dual Facets of Cache Benefits

Performance Improvement

Cache Capacity

Improvements from Write Optimizations (clustering) and Read hits

Buffer Pool Benefits

Make sure that the majority of writes can be immediately posted to the cache
Write-back Cache

- **Operation**
  - Receive data from host, return completion
  - Write to disk “later…”

<table>
<thead>
<tr>
<th>Without write-back cache</th>
<th>With write-back cache</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CPU Processing</strong></td>
<td><strong>CPU Processing</strong></td>
</tr>
<tr>
<td><strong>Seek</strong></td>
<td><strong>Cache</strong></td>
</tr>
<tr>
<td><strong>Rotate</strong></td>
<td><strong>Data</strong></td>
</tr>
<tr>
<td><strong>Xfr</strong></td>
<td><strong>CPU Processing</strong></td>
</tr>
<tr>
<td><strong>Time Saved</strong></td>
<td></td>
</tr>
</tbody>
</table>

Done in background

- Seek
- Rotate
- Xfr
Write-back Cache Benefits

- Electronic access speeds
- Reduces disk utilization
  - Improves response time
- Potential optimizations
  - Write aggregation
  - Write overlays
  - Allows reordering
  - “Speed matching” through de-stage algorithms
- Should be non-volatile
  - Mirrored cache an additional availability plus
Cached Performance

Response Time (ms)

Requests Per Second

No Cache
Read Cache
Writeback Cache

SNIA Emerald™ Training ~ February – March 2018
www.sniaemerald.com
Disk Service Times

- **SCSI Ctrl**
- **Storage Ctrl**
- **8KB Xfr** (Same For F/C Drives)
- **Rotate**
- **Seek**
- **ULTRA-SCSI BUS** (1/2 For F/C Bus)

- **Time (ms)**
  - 0
  - 1
  - 2
  - 3
  - 4
  - 5
  - 6
  - 7
  - 8
  - 9
  - 10

- **RPMs**
  - 15K RPM
  - 10K RPM
  - 7200 RPM
  - 5400 RPM
  - 7200 RPM

- **Bus**
  - Ultra-SCSI Bus (1/2 For F/C Bus)
Database Workload Example 10k & 15k RPM

Database Workload
VRaid 1

Response Time (ms)

Requests Per Second

15K RPM
10K RPM
Response Time

- Response time related to utilization

\[ t_r = \frac{t_s}{(1 - u)} \]

- Two ways to reduce response time:
  - Reduce service time
  - Reduce utilization -> queue length reduction
    - Service time reduction
    - Request rate reduction
Improving Performance

- **Goal is to reduce utilization**
  - Must reduce request rate or service time
- **Reducing device request rate:**
  - Service more requests out of cache
    - Increase server cache hit ratio (bigger buffer cache)
    - Array technology (bigger front end array cache)
  - More spindles to reduce per spindle request rate
- **Reducing service time:**
  - Increase cache
  - Use quicker “devices” RPM, more spindles
  - Introduction of Tiered Storage (AO, Smart Tier)
Improving Performance (Cont’d)

- Minimize Disk I/O

- Spread load (striping or partitioning) over
  - Front end Host Ports/Processors
  - Back end Ports/Processors
  - Disk Spindles

- For Sequential Streams, Large Stripes are Better: (X of transfer size)
  - Be aware of the potential of multiple sequential streams -> Random
    - Some Systems are quite good at Minimizing Cache Pollution
Q & A - Storage Performance and IO Load Basics

Thank You for Your Attention

Chuck Paridon
Member SNIA GSI