Demystifying Storage Configurations for Optimal Performance

Steven Johnson
Performance Scientist
Sun Microsystems, Broomfield CO
Agenda

- Overview of Disk Drive Performance
- Compare FC/SAS drives to SATA drives
- Optimal Array configurations for throughput and response times
  - Understanding the Impact of Stripe Depth
    - RAID 1
    - RAID 5
Overview of Disk Drive Performance

I promise to be quick!
Internals to a disk drive

- Platters
- Read/write Heads
- Actuator Arm
Internals to a disk drive

**Latency** – Time for the data to fly under the head

**Seek** - Move the Actuator arm

Outer Diameter (OD)

Inner Diameter (ID)
The next generation of drives

- SATA – approaching 1 TB capacities
  - Very inexpensive, very high capacity, slow
- 3.5 inch drives – Now 500 GBs
  - 10 and 15K versions, good capacity and transfer characteristics. Excellent performance
- 2.5 inch drives – currently largest is 146 GB
  - Lower capacity, very fast access, slightly less transfer rates compared to 3.5 inch.
Sustained Data rates

Sustained MB/sec

MB/Second

SATA 7200 RPM
SAS 2.5 10K RPM
FC 3.5 10K RPM
SAS 2.5 15K RPM
SAS/FC 3.5 15K RPM

© 2008 Storage Developer Conference . All Rights Reserved.
Seek times

Various drive Random read seek times

<table>
<thead>
<tr>
<th>Drive Type</th>
<th>RPM</th>
<th>Time in ms</th>
<th>+31%</th>
<th>+34%</th>
</tr>
</thead>
<tbody>
<tr>
<td>SATA 7200</td>
<td>10K</td>
<td>8.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SAS 2.5 10K</td>
<td>10K</td>
<td>3.8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FC 3.5 10K</td>
<td>10K</td>
<td>4.7</td>
<td></td>
<td>+31%</td>
</tr>
<tr>
<td>SAS 2.5 15K</td>
<td>15K</td>
<td>2.9</td>
<td></td>
<td>+34%</td>
</tr>
<tr>
<td>SAS/FC 3.5</td>
<td>15K</td>
<td>-3.5</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Drive Latencies

Various drive latency

<table>
<thead>
<tr>
<th>Type</th>
<th>Latency (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SATA 7200RPM</td>
<td>4.17</td>
</tr>
<tr>
<td>SAS 2.5 10K</td>
<td>3</td>
</tr>
<tr>
<td>FC 3.5 10K</td>
<td>3</td>
</tr>
<tr>
<td>SAS 2.5 15K</td>
<td>2</td>
</tr>
<tr>
<td>SAS/FC 3.5 15K</td>
<td>2</td>
</tr>
</tbody>
</table>
Break down of a 32K read

Total Service time for a Random read 32K

<table>
<thead>
<tr>
<th>Condition</th>
<th>SATA 7200 RPM</th>
<th>SAS 2.5 10K RPM</th>
<th>FC 3.5 10K RPM</th>
<th>SAS 2.5 15K RPM</th>
<th>SAS/FC 3.5 15K RPM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unproductive</td>
<td>97%</td>
<td>94%</td>
<td>93%</td>
<td>95%</td>
<td>95%</td>
</tr>
<tr>
<td>Productive (GOOD)</td>
<td>97%</td>
<td>94%</td>
<td>93%</td>
<td>95%</td>
<td>95%</td>
</tr>
<tr>
<td>Over Head (BAD)</td>
<td>97%</td>
<td>94%</td>
<td>93%</td>
<td>95%</td>
<td>95%</td>
</tr>
</tbody>
</table>
Drive Service times for 16K Random Read

<table>
<thead>
<tr>
<th>Drive Type</th>
<th>RPM</th>
<th>Time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SATA 7200</td>
<td>7200RPM</td>
<td>12.88</td>
</tr>
<tr>
<td>SAS 2.5 10K</td>
<td>10K RPM</td>
<td>6.98</td>
</tr>
<tr>
<td>FC 3.5 10K</td>
<td>10K RPM</td>
<td>7.90</td>
</tr>
<tr>
<td>SAS 2.5 15K</td>
<td>15K RPM</td>
<td>5.04</td>
</tr>
<tr>
<td>SAS/FC 3.5</td>
<td>15K RPM</td>
<td>5.63</td>
</tr>
</tbody>
</table>

- SATA 7200: 12.88 ms
- SAS 2.5 10K: 6.98 ms (+38%)
- FC 3.5 10K: 7.90 ms (+40%)
- SAS 2.5 15K: 5.04 ms
- SAS/FC 3.5: 5.63 ms
Theoretical I/Os/Second vs Transfer size

Max single threaded Read IOPS

- SAS 2.5 15K RPM
- SAS/FC 3.5 15K RPM
- SAS 2.5 10K RPM
- FC 3.5 10K RPM
- SATA 7200 RPM
Theoretical Drive Transfer speed vs Transfer size

Max single threaded read MB/sec

Transfer size in K

MB/second

- SAS/FC 3.5 15K RPM
- SAS 2.5 15K RPM
- SAS 2.5 10K RPM
- FC 3.5 10K RPM
- SATA 7200 RPM
Increased Queue depth per drive improves throughput

Random Read Performance

- 3.5" 15K SAS
- 2.5" 10K SAVVIO
- SATA

+92% increase in throughput
Optimal Array Configurations

Real Lab Data (but not a guarantee of performance)
Which is more efficient?

1 person with a Shovel?

10 people with a Spoon?

To move a pile of dirt
Many performance problems start with Array Striping

Folk lore states “thin wide stripes” provides optimal performance

 Older storage arrays had 16K default stripe size

Such small stripe sizes kill our performance

WHY?
Write Coalescing

- Most real workloads have “locality”
- Tend to read or write is the same general area
- Write back cache delay writes as much as possible
- Collect all the write data over time, then write it to disk in one IO
- Minimize the write penalty by effectively amortizing seek and latencies over more IOs
RAID 1 (Mirroring)

- Personally I believe there is only one choice: Go BIG! Assuming controller does partial stripe staging.
- Controller write cache will perform write coalescing
- As the data ages out, it will write to disk fairly efficiently
Solutions to the Raid 5 Penalty

- What is the Raid 5 Penalty?
- What is Parity on the fly
- What is Partial Stripe Update
- How parallelism hurts read/write performance
- Real lab data
- Why do we care? ~85% of customer configurations are RAID 5
The Raid 5 Write Penalty

- To do a random write, the array controller must do two reads and two writes for every block written.
- Must read Old Data and Old Parity
- Then XOR out old data, XOR in new data
- Write New Data and New Parity
- Puts 4 times as much load on the system
- Write cache is designed to mask away this delay
Raid 5 update write

- Host Write

Time line:
- New Block 32K
- Xor out old Data
- Xor in new Data
- New Block
- New Parity

Read:
- Old data

Write:
- Old Par
- Parity

Total cost on Storage:
- 2 Reads
- 2 Writes
Raid 5 and “Parity on the Fly”

- Term used for when we have a full stripe in write cache
- Able to calculate Parity with ZERO READS!
- Effectively no penalty, just a little more overhead
Raid 5 “Parity on the fly”

Time line

Host Write

New Block 32K

New Block 32K

New Block 32K

Create New Parity

New Block

New Block

New Block

New Parity

Old data

Old data

Old data

Old Par

Total cost on Storage

0 Reads

4 Writes

32K

32K

32K

Parity

© 2008 Storage Developer Conference . All Rights Reserved.
Partial Stripe Update

- Current intelligent controllers now perform “partial stripe update”
- Calculates what is more efficient
  - Standard parity update (N reads and N writes)
  - Or fill out the stripe with missing data and do a “parity on the fly”
- All is designed to make Raid 5 more efficient
Raid 5 Partial Stripe

Time line

Host Write

- New Block 32K
- New Block 32K
- New Block 16K
- Missing Block 16K

Create New Parity

- New Block 32K
- New Block 32K
- New Block 32K
- New Parity

Old data

- 32K
- 32K
- 16K

Old Par

Total cost on Storage
1 Reads
4 Writes
Disk Drive Speed

- We have all heard of Moore’s Law
  - Silicon density doubles every 12 months
- Similar behavior with Storage Capacity
  - Near doubling in capacity every 12-18 months
  - That increased density increases MB/sec
- Problem is mechanical access times are not keeping up
  - Access times only improve 7-8% each year
Drive efficiency Improves with Larger Transfer sizes

15K RPM random read service time

- Xfer time
- Latency
- Seek Tm

Non Productive time on the Disk Drive

<table>
<thead>
<tr>
<th>Xfer size in K</th>
<th>16</th>
<th>32</th>
<th>64</th>
<th>128</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time in ms</td>
<td>3.5</td>
<td>3.5</td>
<td>3.5</td>
<td>3.5</td>
</tr>
</tbody>
</table>

© 2008 Storage Developer Conference . All Rights Reserved.
Efficiencies on drives

- By transferring more data per seek and latency will improve drive MB/second
- But why should we care?
- It’s being done in Parallel – Its’ going faster right?
- No – it is actually going slower
- When we read from more than one drive, we are no longer working with averages!!
Each IO is not the same

Must wait for the longer of two IOs.
Each IO is not the same

Must wait for the longer of four IOs.
Best Case single threaded read

R5 Single threaded Response time SE6130

Response time in ms

Xfer size in K

- Slowest of 2 drives
- Slowest of 4 drives
- Slowest of 8 drives

+30%
+65%
Stripe Depth Study

- Stripe depth on the **ST2530**
- Varied from 64K, 128K, and 512K on a **5+1 Raid 5** parity group
- Random reads and Random writes across a wide range of transfer sizes (512 bytes to 1 MB)
- Let's look at MAX IOPS from deep stripes
Only 3 threads against 6 drives Random Read MAX Throughput in IOPS

Max IOPS 3 threads random read

- 0% Delta
- -30% Delta
- -21% Delta

© 2008 Storage Developer Conference. All Rights Reserved.
Random Writes MAX Throughput in IOPS

Max IOPS 3 threads Random Write

-12% Delta
- 49% Delta
- 17% Delta

© 2008 Storage Developer Conference . All Rights Reserved.
Mixed workloads 50% Write

Max IOPS 3 threads 50% read random workload

- 2% Delta
- 35% Delta
- 21% Delta

© 2008 Storage Developer Conference. All Rights Reserved.
Read Response Time Improve

Random read 3 threads

Response time in ms.

Xfer size in K

© 2008 Storage Developer Conference . All Rights Reserved.
Better Write Response Times

Random Write 3 threads

Response time in ms vs. Xfer size in K

- 64k_3th
- 128k_3th
- 512k_3th

© 2008 Storage Developer Conference . All Rights Reserved.
Skip Sequential Workloads

100% Write Skip Sequential

IOPS

0.5K------------------------1M

0.5K------------------------1M

25% seek

50% seek

15% seek

10% seek

64K

128K

512K

test case

49

61

73

85
Lotus Notes workload

Lotus Notes workload 25K xfer size / 38% Read
6540 3+1 RAID 5

Source: Sun StorageTek 6540/6140 Tuning and sizing guide for Lotus Notes/Domino environments by Tom Hanvey
In Summary

- Overview of SAS, FC, SATA
  - SATA is for very infrequently access data or very very sequential workloads
  - Transfer times are dropping rapidly with Moore’s Law
- Stripe Depth Matters!
  - Breaking a system read or write into multiple IOs across multiple drives hurts Response Time and Throughput
Questions?

Steven Johnson
Performance Scientist
Sun Microsystems