Measuring Enterprise SSD Performance

Doug Rollins
Micron Technology
DougRollins@Micron.com
Agenda

- Goals
- Host setup
- Common tools
- Mechanics of a write-saturation plot
- Example
GOALS…
Common Test Goals

- **Accuracy**—minimize the impact of:
  - Host (platform, OS, application)—
    - Service packs, driver revisions, firmware revisions, cache settings, patch levels, kernel levels, BIOS settings, target options, etc…
  - Target uniqueness—
    - OK to investigate/exploit a target’s favorable/unfavorable tendencies
    - Be aware of them and be “fair”
  - Test variables

- **Consistency:**
  - Test process = deterministic (same inputs=same outputs)
  - Congruent to application

- **Repeatability:**
  - “Your” results = “my” results
Common Assumptions: Enterprise

- **Enterprise assumptions:**
  - Drive is always full
  - Drive is always being accessed
  - Decisions are made on steady state performance, typical usage model
  - Steady state ≠ full drive ≠ worst case
  - Operate the drive in a manner consistent with common enterprise use

- **Fill state and access assumptions:**

<table>
<thead>
<tr>
<th>Fill State</th>
<th>Access Interval</th>
<th>Downtime Accepted</th>
<th>Failure Consequences</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enterprise</td>
<td>Full</td>
<td>24x7</td>
<td>None</td>
</tr>
<tr>
<td>Client</td>
<td>Not full</td>
<td>&lt; 8x5</td>
<td>Moderate</td>
</tr>
</tbody>
</table>

- **Decision criteria = typical usage model**
  - Database application => good random R/W performance
  - Video serving => good large block R/W performance
Common Assumptions: Enterprise

- **Enterprise assumptions:**
  - Drive is always full
  - Drive is always being accessed
  - Decisions are made on steady state performance, typical usage model
  - Steady state ≠ full drive ≠ worst case
  - Operate the drive in a manner consistent with common enterprise use

- **Steady state defined (from SNIA PTS):**
  - \( \text{Max}(y) - \text{Min}(y) \) within the measurement window is no more than 20% of the Ave\( (y) \) within the measurement window AND
  - \([\text{Max}(y) \text{ as defined by the linear curve fit of the data with the measurement window}] - [\text{Max}(y) \text{ as defined by the linear curve fit of the data with the measurement window}]\) is within 10% of the average within the measurement window

- **Full drive defined:** Drive has been written over some multiple (could be 1X) of the user-accessible LBA space by a fixed pattern that may be invariant from the test stimulus (i.e. 2X user LBA space written with 128K SEQ WRI)

- **Worst case defined:** Drive has been stimulated over some fixed time with a workload *intentionally designed* to demonstrate the drive’s worst possible performance. Typically this includes:
  - Small transfers mixed with large transfers
  - Intentionally misaligned writes
Common Assumptions: Enterprise

- **Test as the drive would be used:**
  - No power backup/holdup on drive => disable write buffer
  - Focus on the relevant metrics—
    - Highly or lightly loaded queue?
    - Single-threaded IO?
    - Corner cases are interesting, but ensure there is a (relevant) point for their measurement (not just trying to break a drive)

**Congruence with actual usage model is paramount**
Test Flow: Overview

- Purge
- Precondition
- Test
- Collect & Report
Test Flow: Overview

- Determine time granularity required for test

- Set the distance between x-axis (time) marks:
  - Small duration = granular results, but may interfere with actual host I/O!
  - Large duration = less granular, may miss key state transitions and/or latency data

- Determine duration to ensure steady state is reached:
  - Empirically
  - Don’t assume all drives behave the same!
HOST SETUP…
Example: Server 2008

To ensure run-to-run consistency, settle on a patch/SP level, then disable the OS from installing updates (this also prevents rebooting in the middle of a test run)…
Example: Server 2008

Ensure that the drive is recognized...
Example: Server 2008

If the drive does not support power backup/holdup, disable the write buffer…

Clear this check mark
Example: Server 2008

Launch IOMeter...
Example: Server 2008

Set up the precondition: 128K Sequential Write over 2X the user capacity, set alignment...
Q: How do I know when I’ve written 2X the user capacity? IOMeter 2006 write duration is set by time?
Q: How do I know when I’ve written 2X the user capacity? IOMeter 2006 write duration is set by time?

A: Empirically
Example: Server 2008

Set up the precondition “dwell time” and queue depth (good rule of thumb – start with 5 minutes, queue = device max)…

The “run time” will set the granularity of the results (distance between x-axis marks) in the results plot.

For the precondition, we want the target to run as fast as possible, so load the queue!!!
Set up the rest of the run...

The number of times the access spec is added = the number of x-axis intervals (for a drive you’ve never tested, more is better); click Add once for each 5-minute run you want to add to the precondition.

Add the Precondition access specification.

Add only the Precondition access specification!
Example: Server 2008

Run the test and the examine results; the compiled output should look something like this:
Example: Server 2008

Look at the MB/s vs. the time graph and determine the write rate (in this case, about 300 MB/s)…

100GB SLC Precondition: FOB to Steady State

![Graph showing MB/s vs iteration number for 100GB SLC precondition from FOB to Steady State. The graph indicates a steady write rate of about 300 MB/s.]
Example: Server 2008

Calculate the time to fill the drive twice, based on this write rate:

<table>
<thead>
<tr>
<th>Drive capacity:</th>
<th>100GB = 102400MB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Write rate:</td>
<td>300MB/s</td>
</tr>
<tr>
<td>Minutes to fill (1x):</td>
<td>6</td>
</tr>
<tr>
<td>Minutes to fill (2x):</td>
<td>12</td>
</tr>
<tr>
<td>Precondition time:</td>
<td>12 minutes</td>
</tr>
</tbody>
</table>

**Note:** This is the time for the drive to go from a purged state into a preconditioned state. This can be different for different drives, different media, and different firmware revisions. It is constant for each DUT, however!
Now that the drive is preconditioned, test it...
Example: Server 2008

Launch IOMeter...
Example: Server 2008

Set up the test—for example, 4K Random Write...
Example: Server 2008

Set up the precondition “dwell time” and queue depth (good rule of thumb—start with 5 minutes, queue = device max)...

The “run time” will set the granularity of the results (distance between x-axis marks) in the results plot.

For the precondition, we want the target to run as fast as possible, so load the queue!!!
Set up the rest of the run and go...

From experience, this 100GB drive is always in steady state by the 25th iteration, so add the access specification 25 times.

Add the 4K Random Write access specification.

Add only the 4K Random Write specification!
**Example: SLC SATA SSD Write Saturation**

### Steady State Region

<table>
<thead>
<tr>
<th>Iteration Number</th>
<th>Transfer Size</th>
<th>% Read</th>
<th>% Random</th>
<th>IOPs</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4096</td>
<td>0</td>
<td>100</td>
<td>35460.98405</td>
</tr>
<tr>
<td>2</td>
<td>4096</td>
<td>0</td>
<td>100</td>
<td>17382.48189</td>
</tr>
<tr>
<td>3</td>
<td>4096</td>
<td>0</td>
<td>100</td>
<td>20179.53067</td>
</tr>
<tr>
<td>4</td>
<td>4096</td>
<td>0</td>
<td>100</td>
<td>19460.9479</td>
</tr>
<tr>
<td>5</td>
<td>4096</td>
<td>0</td>
<td>100</td>
<td>19568.64219</td>
</tr>
<tr>
<td>6</td>
<td>4096</td>
<td>0</td>
<td>100</td>
<td>19657.92814</td>
</tr>
<tr>
<td>7</td>
<td>4096</td>
<td>0</td>
<td>100</td>
<td>19573.7974</td>
</tr>
<tr>
<td>8</td>
<td>4096</td>
<td>0</td>
<td>100</td>
<td>19561.21846</td>
</tr>
<tr>
<td>9</td>
<td>4096</td>
<td>0</td>
<td>100</td>
<td>19572.46237</td>
</tr>
<tr>
<td>10</td>
<td>4096</td>
<td>0</td>
<td>100</td>
<td>19597.42278</td>
</tr>
<tr>
<td>11</td>
<td>4096</td>
<td>0</td>
<td>100</td>
<td>19577.95913</td>
</tr>
<tr>
<td>12</td>
<td>4096</td>
<td>0</td>
<td>100</td>
<td>19595.8534</td>
</tr>
<tr>
<td>13</td>
<td>4096</td>
<td>0</td>
<td>100</td>
<td>19442.13415</td>
</tr>
<tr>
<td>14</td>
<td>4096</td>
<td>0</td>
<td>100</td>
<td>18986.06345</td>
</tr>
<tr>
<td>15</td>
<td>4096</td>
<td>0</td>
<td>100</td>
<td>19065.45231</td>
</tr>
<tr>
<td>16</td>
<td>4096</td>
<td>0</td>
<td>100</td>
<td>18993.24672</td>
</tr>
<tr>
<td>17</td>
<td>4096</td>
<td>0</td>
<td>100</td>
<td>19016.87946</td>
</tr>
<tr>
<td>18</td>
<td>4096</td>
<td>0</td>
<td>100</td>
<td>19011.96523</td>
</tr>
<tr>
<td>19</td>
<td>4096</td>
<td>0</td>
<td>100</td>
<td>19024.61239</td>
</tr>
<tr>
<td>20</td>
<td>4096</td>
<td>0</td>
<td>100</td>
<td>19045.18973</td>
</tr>
<tr>
<td>21</td>
<td>4096</td>
<td>0</td>
<td>100</td>
<td>18990.85739</td>
</tr>
<tr>
<td>22</td>
<td>4096</td>
<td>0</td>
<td>100</td>
<td>18997.49995</td>
</tr>
</tbody>
</table>

Flash Memory Summit 2011
Santa Clara, CA
1. Assume drive is always full and always under maximum load
2. Assume steady state is region of interest
3. Always start from a repeatable, known, fixed point
4. Always precondition the drive in the same way
5. Always stimulate the drive with a single fixed stimulus until steady state is reached, then purge and repeat for other stimuli*

*Reads are an exception; all read measurements may be done at the same time, just after precondition
Conclusions

Thank you!

Q&A?

Revisit Micron’s FMS 2011 presentations at:
www.micron.com/fms