Design Tradeoffs for SSD Performance

Nitin Agrawal*, Vijayan Prabhakaran, Ted Wobber, John D. Davis, Mark Manasse, Rina Panigrahy

Microsoft Research, Silicon Valley
*University of Wisconsin-Madison
Outline

- Introduction
- SSD Architecture
- Simulation Techniques
- Experiment
- Conclusions
Introduction (1/2)

- The advent of NAND-flash based SSD is a sea change in the architecture of computer storage subsystem.

- Compared to traditional disks, SSDs have lower access time and latency.

- But the SSD lifetime depend on erasing times.
Introduction (2/2)

- The following systems issues are relevant to SSD performance:
  - Data placement
    - It’s critical to load balancing
  - Parallelism
    - Operating in parallel can improve the efficiency
  - Write ordering
    - Random or Sequential writing
  - Workload management
    - Performance is highly workload-dependent
SSD Architecture (1/2)

SSD Logical Components
SSD Architecture (2/2)

Flash internals
Simulation Techniques

- Interleaving
- Ganging
  - Shared bus gang
  - Shared Control gang
- Cleaning
  - Overprovisioning
  - Copy-back
  - Wear-leveling
- Striping
Interleaved page copying

Source Plane 0
Dest Plane 0
Source Plane 1
Dest Plane 1
Source Plane 2
Dest Plane 2
Source Plane 3
Dest Plane 3

Time →

Read
Xfer
Write
Ganging

- Shared bus gang

- Shared control gang
Cleaning

- The following things are important for SSD
  - Reduce the number of times of erasing
    - Overprovisioning
  - Improve the erasing efficiency
    - Copy-back
  - Erase blocks averagely
    - Wear-leveling
Over-provisioning

- It provides extra memory capacity

- The SSD controller uses these extra cells to create pre-erased blocks
Copy-back (1/2)

- Cleaning a block involves moving any valid pages to another block.

- If the source and destination blocks are in the same plane, pages can be moved using copy-back feature.

- Pages can be moved without transferring across the serial pins.
Copy-back (2/2)
Wear-leveling (1/2)

- `ageVariance` (say 20%)

- `retirementAge` (say 85% of the average remaining lifetime)
Wear-leveling (2/2)

- Greedy approach
  - The blocks which contain hot data might be modified frequently

- Rate-limit
  - Eg. Drop a block’s remaining lifetime from 80% to 0%

- Migrate
  - If the chosen block’s remaining lifetime is below \textit{retirementAge} of the remaining lifetime, we can migrate cold data into this block
Striping

- Data striping is the technique of segmenting logically sequential data.
- By performing segment accesses on multiple devices, multiple segments can be accessed concurrently.
- This provides more data access throughput, which avoids causing the processor to idly wait for data accesses.
Simulation results

- Impact of interleaving
- Shared-control ganging
- Shared-bus ganging performance
- Cleaning frequency and efficiency
- Impact of minimum free blocks
- Block Wear
Impact of interleaving

Performance Improvement with Interleaving

Average Queue Length

IO/s

# requests

TPC-C  Iozone  Postmark  Exchange

None  Dies  Plane-pairs
Shared-control ganging

![Graph showing response time for different ganging configurations for TPC-C, lozone, Postmark, and Exchange. The graph compares 8-way, 4-way, 2-way, Sync 8-way, Sync 4-way, and Sync 2-way ganging.]
### Shared-bus gang performance

<table>
<thead>
<tr>
<th></th>
<th>No gang</th>
<th>8-gang</th>
<th>16-gang</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host IO Latency</td>
<td>237 μs</td>
<td>553 μs</td>
<td>746 μs</td>
</tr>
<tr>
<td>IOPS per gang</td>
<td>4425</td>
<td>1807</td>
<td>1340</td>
</tr>
</tbody>
</table>
Cleaning frequency and efficiency

<table>
<thead>
<tr>
<th></th>
<th># cleaned</th>
<th>Avg. time (ms)</th>
<th>Efficiency</th>
</tr>
</thead>
<tbody>
<tr>
<td>TPC-C (inter-plane)</td>
<td>114</td>
<td>9.65</td>
<td>70%</td>
</tr>
<tr>
<td>TPC-C (copy-back)</td>
<td>108</td>
<td>5.85</td>
<td>70%</td>
</tr>
<tr>
<td>IOzone</td>
<td>101170</td>
<td>1.5</td>
<td>100%</td>
</tr>
<tr>
<td>Postmark</td>
<td>2693</td>
<td>1.5</td>
<td>100%</td>
</tr>
</tbody>
</table>
Impact of minimum free blocks

(a) Access latency vs. minimum free blocks before cleaning starts (%)

(b) Pages moved during cleaning

Pages moved (in 1000s)
## Block Wear

<table>
<thead>
<tr>
<th></th>
<th>Mean Lifetime</th>
<th>Std.Dev.</th>
<th>Expired blocks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Greedy</td>
<td>43.82</td>
<td>13.47</td>
<td>223</td>
</tr>
<tr>
<td>+ Rate-limiting</td>
<td>43.82</td>
<td>13.42</td>
<td>153</td>
</tr>
<tr>
<td>+ Migration</td>
<td>43.34</td>
<td>5.71</td>
<td>0</td>
</tr>
</tbody>
</table>
## Tradeoff Summary

<table>
<thead>
<tr>
<th>Technique</th>
<th>Positives</th>
<th>Negatives</th>
</tr>
</thead>
<tbody>
<tr>
<td>Large allocation pool</td>
<td>Load balancing</td>
<td>Few intra-chip ops</td>
</tr>
<tr>
<td>Large page size</td>
<td>Small page table</td>
<td>Read-modify-writes</td>
</tr>
<tr>
<td>Overprovisioning</td>
<td>Less cleaning</td>
<td>Reduced capacity</td>
</tr>
<tr>
<td>Ganging</td>
<td>Sparser wiring</td>
<td>Reduced parallelism</td>
</tr>
<tr>
<td>Striping</td>
<td>Concurrency</td>
<td>Loss of locality</td>
</tr>
</tbody>
</table>
Conclusions

- We have shown there are numerous design tradeoffs for SSDs that impact performance.

- The block-access nature of NAND suggests that a block-oriented interface will often be appropriate.

- Our work represents a step towards understanding and optimizing the performance.