Solid State Drives: Use, Performance, Caching, and More

VeloBit
Faster Storage Made Easy

September 12, 2012

Agenda

- Solid State Drive technology overview
- SSD performance characteristics
- Durability and Reliability of SSDs
- Deploying SSDs in your environment
- Using SSDs for caching

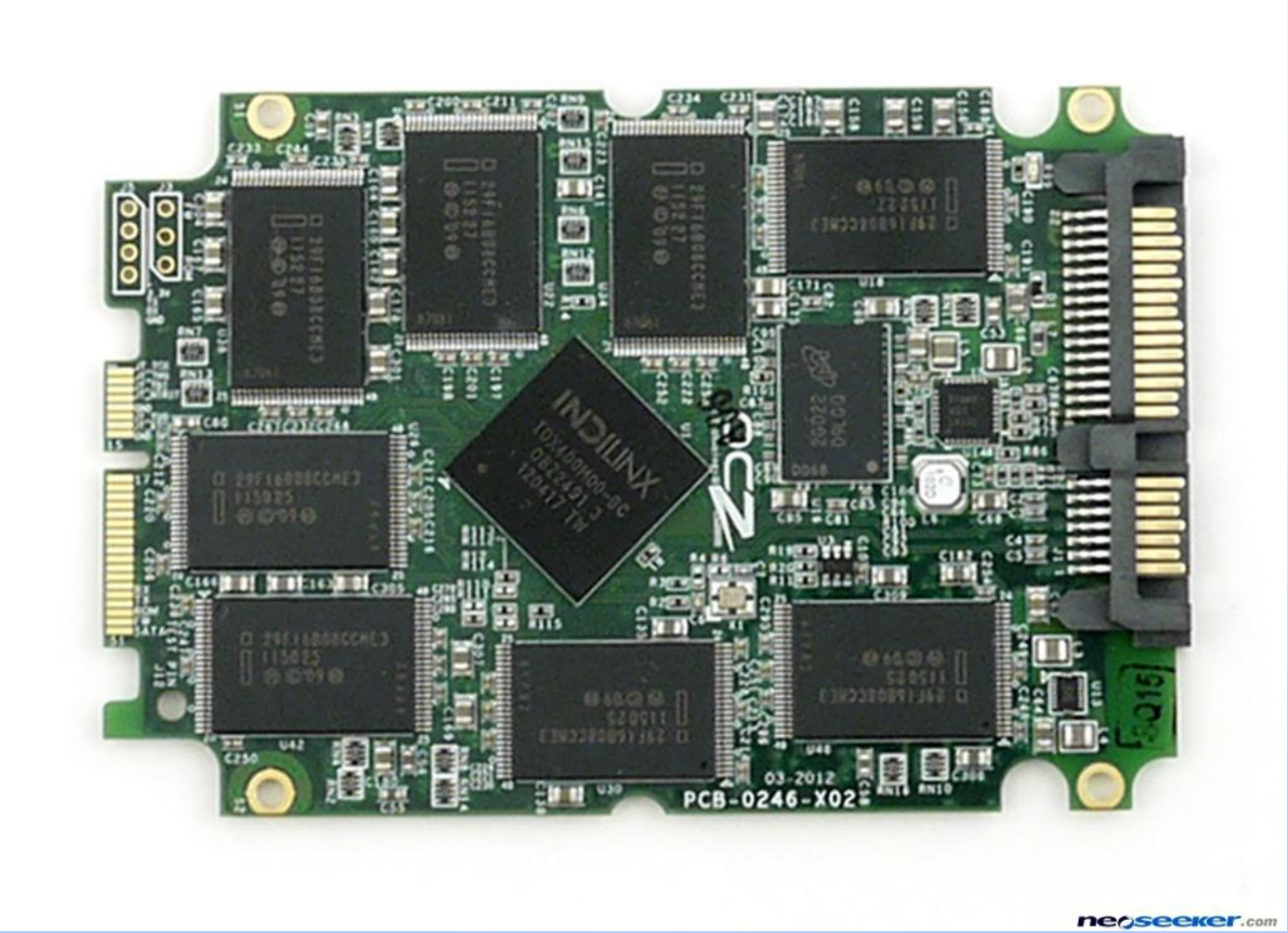
Solid State Drive Overview

- Uses flash memory instead of spinning magnetic media for system nonvolatile (“disk”) storage
- Form factors:
  - **Drive form factor (2.5” or 3.5” SATA and SAS)**
  - **PCI Express card**
- Most NAND flash chips have nearly identical characteristics
  - **MLC vs. SLC**
- SSD Firmware is critically important to drive operation and is the differentiator between different SSDs
  - **Performance and durability characteristics vary widely**
- Most operate with internal 4096 byte blocks

Technology Timeline

- 1995 – First flash-based solid state drive M-Systems
- 2007 – Fusion-io introduces a PCI Express with 100k IOPS
- 2007 – Dell ships ultraportable laptop with SanDisk SSD
- 2008 – Apple release MacBook Air with 64GB SSD
- 2008 – EMC Corp. ships “enterprise” SSDs
- 2009 – OCZ demonstrates 1TB PCIe flash
- 2010 – MacBook Air ships with SSD standard
- 2010 – Fusion-io releases 5.12 TB PCIe drive
- 2012 – OCZ releases R4 CloudServ PCIe with transfer speeds of 6.5 GB/s and 1.4 million IOPS

What's inside?



SSD Flash Layout

- Page: 4096 bytes
  - **Smallest granularity for reads and (new) writes**
  - **Fast read: ~25 µs**
  - **Slower to program: ~300 µs**
  - **Cannot be rewritten**
- Block: 128KB – 512KB
  - **Smallest granularity for erase**
  - **Much slower to erase: ~2 ms**
- Each page contains extra space for ECC and other information used by the drive firmware

SSD Firmware

- Flash Translation Layer
  - **Translates LBA -> Flash Physical Block Address**
  - **Writes proceed sequentially**
  - **Rewrites must be remapped**
  - **This requires “garbage collection”**
- Mapping traditional 512 byte sectors to 4096 byte blocks
  - **Alignment issue**
- Wear Leveling
  - **Limited number of rewrites for a cell of flash: 10^5 to 10^6 cycles**
- Error detection and correction
- Compression

Write Amplification

- Due to garbage collection a user page write will result in more than one page actually being written on average
- Always >1
- Depends on the user workload
- No issue for reads
- Flash storage is “over provisioned” by the manufacturer
  - **This capacity is hidden to the user**
  - **Write performance gets worse when the drive fills up due to increased garbage collection requirements**
  - **Drive firmware compression can help (if your data is compressible!)**

SSD Performance

- Much better than spinning disks for:
  - **Random reads**
  - **Random writes**
  - **Fragmented data**
- Spinning disks can be better at:
  - **Entirely sequential reads and writes**
  - **Same location on disk modified constantly**
- Factors that affect SSD Performance:
  - **Read/write ratio**
  - **Utilized capacity and over provisioning**
  - **Compressibility**

When Sequential goes Random

- Sometimes what you think is sequential ends up as random IO
  - **Virtualization: Many servers accessing the same resource**
  - **Fragmentation**
  - **Multiple users consuming sequential streams**

Enterprise vs. Consumer Drives

- Consumer grade drives are optimized for a read heavy workload
  - **Less overprovisioning**
  - **Firmware engineering**
  - **All MLC**
- Enterprise grade drives are optimized for mixed workload
  - **Up to 3x overprovisioned**
  - **Firmware engineering optimized for writes and overwrites**
  - **More DRAM onboard for cache**
  - **MLC and SLC**

SSD Reliability and Durability

- A write heavy workload is even worse than it seems due to write amplification
- Drive firmware remaps dead cells
  - **This leads to a reduction in over provisioning**
  - **More garbage collection overhead**
  - **Performance suffers**
- Eventually the drive may fail
  - **Lots of cells need to die before this point is reached**
  - **Performance will suffer first**
- Bottom line: write heavy workloads are not ideal for SSDs

Deploying SSDs

- Replace all storage with SSDs
  - **Expensive: SSDs have much higher $$/GB**
  - **Not all workloads are ideal for SSD storage**
  - **Limited capacity**
- Replace some storage with SSDs
  - **Save some money**
  - **Avoid utilizing SSDs in cases where the workload is not ideal**
  - **Keep bulk storage on slower, cheaper, spinning disks**
  - **Tiering is manual, complex, time-consuming, hard to maintain**
- Use SSDs as Cache devices
  - **Software determines what should go on the SSD**
  - **Adapt to workload automatically**
  - **Same performance with less $$$**
  - **Bulk data remains on existing cheap storage: no migration**

SSD as Cache

- Driver automatically stores frequently accessed data on the Solid State Disk
- Avoid the expense of storing all data on pricey SSDs
  - **In most environments on any given day only a fraction of your total data is accessed**
- No need to expend administrative effort in deciding what data should get faster storage
- Leave your current storage environment in place
  - **No storage change**
  - **Non-invasive, snap in solution**
- Write through vs. Write back cache
- Block level vs. file level caching

Workloads for Cache

- Cache friendly workloads
  - Hot spots with repeated access
  - OLTP databases, or analytic databases where a subset is considered
  - Database indexes
  - File system metadata (MFT, inodes, etc)
- Cache unfriendly workloads
  - All data is accessed evenly and is larger than the cache size
  - Broad analytics
  - Data write only or accessed only once

SSD as Cache: Advanced features

- Compression
  - **Better performance: less IO volume**
  - **Better utilization of expensive SSD resources**
- Deduplication
  - **When deployed on a SAN system or Virtualization System there can be a lot of duplicated data!**
  - **Dedupe means better utilization of expensive SSDs**
- Opportunity to handle rewrites better
  - **Use delta compression to reduce the quantity of writes**
  - **Delay writes to SSD using RAM to stage updates so frequent updates get batched**
- Access pattern can be engineered with awareness of flash's limitations

VeloBit

- Block level SSD caching software
  - **Filesystem/data agnostic**
- Works with Linux, Windows, virtualization environments including KVM, Xen, and VMWare
- Advanced features including deduplication and delta compression
- Works with any SSD from consumer to enterprise and even high end PCI Express drives
- Configurable as either a write-through or write-back cache

Conclusions

- SSDs are getting cheaper but are still significantly more expensive per Gigabyte
- SSDs are reliable but write heavy applications can reduce performance and lead to early failure
- Limited capacity means that it will take more drives and administrative effort to replace all storage with SSDs
- Instead of deploying SSDs “manually” consider a caching solution that utilizes SSDs