SSDs – basics & details on performance

Werner Fischer, Technology Specialist Thomas-Krenn.AG

LinuxTag 2011, May 11th - 14th 2011, Berlin / Germany
SSDs – basics & details on performance

Werner Fischer, Technology Specialist Thomas-Krenn.AG
LinuxTag 2011, May 11th - 14th 2011, Berlin / Germany
The last talk before LinuxNacht

7 p.m.
Umspannwerk
Ohlauer Str. 43
Agenda

1) SSD layout

2) Write techniques

3) Usage examples

4) Configurations tips

Source: maximumpc.com
Agenda

1) SSD layout
   • memory cells
   • pages & blocks
   • planes
   • dies
   • TSOPs & SSDs

2) Write techniques

3) Usage examples

4) Configurations tips
1) SSD layout

- **memory cells**
  - NAND memory cell = MOS transistor with floating gate
  - permanently store charge
  - programming puts electrons on floating gate
  - erase takes them off
  - one program/erase (p/e) cycle is a round trip by the electrons
  - back-and-forth round trips gradually damage the tunnel oxide
  - endurance is limited, measured in number of p/e cycles:
    - 50nm MLC ~ 10.000 p/e cycles
    - 34nm/25nm/20nm MLC ~ 3.000 – 5.000 p/e cycles

Source: Intel
1) SSD layout

- **memory cells**
  - SLC (Single Level Cell) → 1 Bit per memory cell
  - MLC (Multi Level Cell) → 2 Bits per memory cell
  - TLC (Triple Level Cell) → 3 Bits per memory cell
  - 16LC (16 Level Cell) → 4 Bits per memory cell

Source: anandtech.com
1) SSD layout

• **pages: multiple memory cells**
  
  – one page is the smallest structure which can be *read* or *written*

• **blocks: multiple pages**

  – one block is the smallest structure which can be *erased*

  – e.g.
    
    one block = 128 pages á 4 KiB (with MLC 16.384 memory cells per page)
    
    → 512 KiB Block

  – newer SSDs (25nm/20nm Intel/Micron or 24nm/19nm Sandisk/Toshiba)
    
    one block = 256 pages á 8 KiB
    
    → 2 MiB Block

Source: anandtech.com
1) SSD layout

• planes
  – multiple blocks make up a plane
  – e.g. 1,024 Blocks = 1 Plane
  – 25nm Intel/Micron: 1 Plane = 2 GiByte

Source: anandtech.com
1) SSD layout

- **dies**
  - multiple planes make up a die, e.g. 4 Planes = 1 Die

Intel/Micron: dies with 64 GiBit (8 GiByte)

Source: http://newsroom.intel.com/community/intel_newsroom/blog/2011/04/14

wafer
Source: Intel/Micron
1) SSD layout

- **TSOPs (thin small outline packages)**
  - multiple dies make up a TSOP
  - typically one – two dies in a TSOP
  - up to eight dies possible
  \[\rightarrow 64 \text{ GiByte in a TSOP}\]

- **SSDs**
  - multiple TSOPs (e.g. ten) make up a SSD
  - currently capacities up to 600 GB
Agenda

1) SSD layout

2) Write techniques
   - spare area
   - wear leveling
   - ATA TRIM
   - garbage collection
   - secure erase
   - endurance

3) Usage examples

4) Configurations tips
2) Write techniques

- **spare area**
  - typically between 7% and 28% of net capacity
  - e.g. 160 GByte visible, but actual capacity is 160 GiByte (171.8 GByte → 11.8 GByte Spare Area)
  - spare area is used for
    - read/modify/write
    - wear leveling
    - bad block replacement
2) Write techniques

- **wear leveling**
  - flash memory cells can only be erased (written) a limited amount of times
  - wear leveling distributes the wearout over all memory cells
2) Write techniques

• **ATA TRIM**
  - OS tells the SSD which LBAs are not needed anymore and can be erased
  - increases the number of deleted blocks, increases the write performance
  - ATA TRIM must be supported by
    • SSD
    • operating system
    • file system
2) Write techniques

- garbage collection
  - at times without I/O the SSD controller merges partly-filled blocks
  - increases the number of deleted blocks
2) Write techniques

- **secure erase**
  - all data gets lost
  - for most SSDs, this deletes all blocks of the SSD by applying an extinction voltage
  - afterwards all blocks are deleted → higher write performance
  - recommended when
    - a used SSD will be used for a different application
    - after performance tests have been done and the SSD should be used for production usage
  - newer SSD with integrated encryption only delete encryption key when doing a secure erase – TRIM is needed there for deleting all blocks
2) Write techniques

- **endurance**
  - **bad blocks**
    - erase slows down with p/e cycles
    - if a NAND block fails to erase, the NAND reports that and the controller will use another block instead
    - no lost data, a failed NAND block is not a problem (as long as there is enough spare capacity)
  - **write data errors**
    - RBER (raw bit error rate) – corrected by ECC
    - RBER gradually increases with p/e cycles
    - ECC used for correction
    - UBER (uncorrectable bit error rate) must be kept very low (<1 error out of every \(10^{15}\) to \(10^{16}\) accesses)
2) Write techniques

- **endurance**
  - data retention
    - number of hours (days/years) how long the data can be written if the device is powered off and not in use
    - ECC can correct a limited number of errors
    - retention time decreases with p/e cycles
  - defects
- **all NAND devices have a „wearout cliff“**
  - new JDEC standards (TBW – Terabytes written)

Source: Intel
Agenda

1) SSD layout

2) Write techniques

3) Usage examples
   - SSD as (small) boot device
   - SSD as replacement for a single HDD
   - SSDs in a RAID configuration
   - SSD as cache

4) Configuration tips
3) Usage examples

- **SSD as (small) boot device**
  - low number of p/e cycles, daily turnover e.g. 0.1x
  - low SSD capacity (e.g. 40 GB or 80 GB) is enough (lower costs)
  - shortened boot-up times
  - programs start faster
  - increases the productivity when working at the PC

80 GB $
3) Usage examples

- **SSD as replacement for a single HDD**
  - normal/low number of p/e cycles, daily turnover e.g. 0,5x; often less
  - middle/higher SSD capacity (middle/higher costs)
  - less power usage and less waste heat, as there is no HDD any more
  - very interesting option for laptops:
    - increases run-time of battery
    - decreases weight
    - increases productivity
3) Usage examples

- **SSDs in a RAID configuration**
  - normal/low number of p/e cycles, daily turnover e.g. 0.5x; often less
  - middle/higher SSD capacity (middle/higher costs)
  - ATA TRIM can not be used with RAID controllers

160-600 GB

$\text{$$$$}$
3) Usage examples

- **SSD as cache**
  - high number of p/e cycles, daily turnover e.g. 10x
  - SSD endurance must be monitored
    - increased spare area increases endurance
  - examples
    - Adaptec maxCache
    - cache device for ZFS

32-600 GB

$\text{$$$$}$
Agenda

1) SSD layout

2) Write techniques

3) Usage examples

4) Configuration tips
   - use AHCI
   - secure erase / full TRIM before production use
   - use ATA TRIM
   - align partitions and file systems
   - use over-provisioning
4) Configuration tips

- use AHCI
  - NCQ (Native Command Queuing)
  - LPM (Link Power Management)
    - use Device Initiated Interface Power Management (DIPM)
- secure erase / full TRIM before production use
  - before doing the partitioning
  - erases all blocks of the SSD
  - increases write performance
- use ATA TRIM
  - Linux 2.6.33 or higher (e.g. Ubuntu 10.10)
  - batched discard support in 2.6.37 (FITRIM ioctl), Ext3 & XFS support batched discard with 2.6.38
4) Configuration tips

- **align partition and file systems**
  - wrong alignment:
  
  ![Diagram showing wrong alignment]
  
  - use fdisk parameters: `fdisk -c -u /dev/sda`
  - correct alignment:

  ![Diagram showing correct alignment]
4) Configuration tips

- over-provisioning (increase spare area)

Source: Intel
Conclusions

technology of SSDs has evolved

prices per GByte decrease (25 nm, later 20nm)

endurance planning possible
with new JEDEC standard

→ SSDs will get even more important in the future
Thanks for your time!

wfischer@thomas-krenn.com
hall 7.2a, booth 143