Garbage Collection

Understanding Foreground vs. Background GC and Other Related Elements

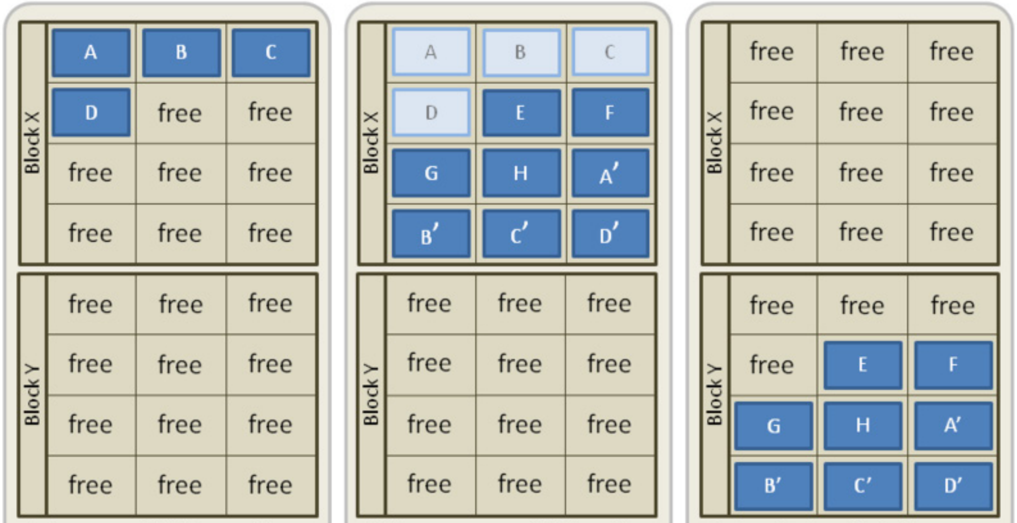
Kent Smith

Sr. Director of Corporate Marketing

SandForce

Understanding Garbage Collection (GC)

- In flash memory, GC is the process of relocating existing data, deleting stale data, and creating empty blocks for new data
- All SSDs will have some form of GC – it is not an optional feature
- NAND flash cannot directly overwrite a page with data; it has to be first erased
- One full block of pages has to be erased, not just one page
- GC starts after each page has been written one time
- Valid data is consolidated and written into new blocks
- Invalid (replaced) data is ignored and gets erased
- Wear leveling mainly occurs during GC



1. Four pages (A-D) are written to a block (X). Individual pages can be written at any time if they are currently free (erased).

2. Four new pages (E-H) and four replacement pages (A′-D′) are written to the block (X). The original A-D pages are now invalid (stale) data, but cannot be overwritten until the whole block is erased.

3. In order to write to the pages with stale data (A-D) all good pages (E-H & A′-D′) are read and written to a new block (Y) then the old block (X) is erased. This last step is *garbage collection*.

Source: Wikipedia

How the OS Deletes Data

- The OS tracks what files are present and what logical blocks are holding the files
- SSDs do not understand the file structure of an OS; they only track valid data locations reported by the OS
- When the OS deletes a file, it marks the file's space in its logical table as free
  - It does not tell the drive anything
- When the OS writes a new file to the drive, it will eventually write to the previously used spaces in the table
  - An SSD only knows data is no longer needed when the OS tells it to write to an address that already contains data

Understanding the TRIM Command

- The OS* sends a TRIM command** at the point of file deletion
- The SSD marks the indicated locations as invalid data

- TRIM Features:
  - Prevents GC on invalid data
  - Increases the free space known to the SSD controller

- TRIM Benefits:
  - Higher throughput – Faster host write speeds because less time writing for GC
  - Improved endurance – Reduced writes to the flash
  - Lower write amplification – Less data rewritten and more free space is available during GC

- TRIM does not generally work behind a RAID environment

* Only more recent Operating Systems (Win 7, Win Server 2008 R2, Linux 2.6.33, FreeBSD 8.2, Open Solaris, Mac OS X Lion) http://en.wikipedia.org/wiki/TRIM
** Linux "discard" parameter

OS File Writes and Deletes without TRIM

	1. User writes four new files	2. User deletes file "C"	3. User writes new file "E"
OS Logical View	File A, File B File C, File D Free	File A, File B File D Free, Free	File A, File B File E, File D Free
SSD Logical View (LBAs)	A1 A2 A3 B1 B2 B3 B4 B5 B6 C1 C2 D1	A1 A2 A3 B1 B2 B3 B4 B5 B6 C1 C2 D1	A1 A2 A3 B1 B2 B3 B4 B5 B6 E1 E2 D1
SSD Physical View **Over Provisioning**	A1 A2 A3 B1 B2 B3 B4 B5 B6 C1 C2 D1	A1 A2 A3 B1 B2 B3 B4 B5 B6 C1 C2 D1	A1 A2 A3 B1 B2 B3 B4 B5 B6 GC GC D1 E1 E2
	SSD writes new data; only SSD knows about OP	Only OS knows location C1 & C2 are no longer valid, but SSD says its valid and keeps rewriting it during GC	OS writes new file to old logical location; SSD marks old physical location ready for GC and file E gets written elsewhere

No TRIM Limits Available Free Space During GC

OS File Writes and Deletes with TRIM

	1. User writes four new files	2. User deletes file "C" and OS sends TRIM	3. User writes new file "E"
OS Logical View	File A, File B File C, File D Free	File A, File B File D Free	File A, File B File E, File D Free
SSD Logical View (LBAs)	A1 A2 A3 B1 B2 B3 B4 B5 B6 C1 C2 D1	A1 A2 A3 B1 B2 B3 B4 B5 B6 D1	A1 A2 A3 B1 B2 B3 B4 B5 B6 E1 E2 D1
SSD Physical View Over Provisioning	A1 A2 A3 B1 B2 B3 B4 B5 B6 C1 C2 D1	A1 A2 A3 B1 B2 B3 B4 B5 B6 GC GC D1	A1 A2 A3 B1 B2 B3 B4 B5 B6 GC GC D1 E1 E2
	SSD writes new data; only SSD knows about OP	TRIM from OS tells SSD to ignore the data in the location previously holding file "C" during GC	OS writes new file to old location; SSD writes file E to another free area

TRIM Makes More Free Space Available During GC

Advantages of DuraWrite™ and TRIM

DuraWrite and TRIM both provide more free space for the SSD during GC

	1. OS w/o TRIM or 2. RAID Environment	1. OS with TRIM and 2. No RAID
SSDs without DuraWrite	Free Space for GC Data Pending Delete Valid User Data	Free Space for GC Free Space for GC after TRIM Valid User Data
SandForce DRIVEN™ SSDs with DuraWrite	Free Space for GC Larger with DuraWrite Data Pending Delete Valid User Data	Free Space for GC Larger with DuraWrite Free Space for GC after TRIM Valid User Data

Free Space for GC = Higher Performance & Lower WA

Standard vs. Background GC

- Background (or idle-time) GC is important when the SSD write speed is slow
  - Will not further reduce low host write speeds
- The problem is all the “soon to be deleted” data is needlessly moved to new blocks and then Trimmed without being read
  - Reduces flash endurance and drive life

- Standard (or foreground) GC is best with high write speeds
  - Only clears/moves data as needed (just-in-time)
  - Will not GC data soon to be deleted
  - Prevents needless endurance reduction

Visit the SandForce Exhibition Booth

World-class reliability, performance, & power efficiency for enterprise, client, and industrial SSD applications

- Visit us at **booth #407** to see our DuraClass™ technology in action with the latest 24nm MLC flash technology and new non-HDD form factors like the JEDEC MO-300

- Stop by to enter our **free drawings** to win one of many different SandForce Driven SSDs from Corsair, Kingspec, Kingston, OCZ, OWC, Patriot Memory & Viking
  - Free drives given away approximately every 30 minutes!

- See other SandForce Driven™ SSDs in our partners' booths

- Visit the many SandForce Trusted™ SSD ecosystem members