ToyDB Project
Performance Comparison of SSD with RAID1

M.S.Krishna Deepak-120050057
Sashank Gondala-120050050

Objectives:

Use PF layer to simulate a Solid State Disk (SSD, ie., Flash) storage system, and compare its performance for a workload consisting of a sequence of random read-write operations with the situation when we use
(a) a normal disk system in RAID 1
(b) a mix of SSD and standard disks where more active tables are kept in SSDs.

Implementing a SSD:

1) In a Flash SSD, data can only be read and written one page at a time. So, to update a page, Flash reads the page into a cache(buffer), changes it, then writes the page back out again
2) Also, data cannot be overwritten. To overwrite memory that has been written already, we have to erase an “entire block” of memory at once; it is then ready to be written again. This method affects the write speed hugely
3) So, we would write the changed page to some other empty page and change the page numbers mapping.
4) While doing this, we would mark the original pages as “stale”. This is done by maintaining an array for each file which maps every page number in the file to an integer 0 or 1, which shows whether the page is stale or not.

Garbage Collection:

1) As the number of such updates increase, the number of stale pages increase which might become a bottleneck. To avoid this, we will have to implement garbage collection. In this process, we would look at all the blocks which have at least one stale page and process them
2) We will have to move the “not stale” pages in a block into a new block and then erase the whole block(Since, we can only erase a block of memory). This completes the garbage collection process
3) This process will have to be performed from time to time to avoid the growth of total number of stale pages
   Eg:- On a sequence of 500 random updates over a file size of 2000 pages, doing a garbage collection after all the updates would result in the number of stale pages to grow to 500. But doing a garbage collection after 250 updates, would reduce the number of maximum stale pages at any time to 250

4) If the user or operating system erases a file (not just remove parts of it), the file will typically be marked for deletion, but the actual contents on the disk are never actually erased. Because of this, the SSD does not know the LBAs that the file previously occupied can be erased, so the SSD will keep garbage collecting them
5) When a file is deleted, the OS sends the TRIM command along with the pages that no longer contain valid data. This informs the SSD that the pages in use can be erased and reused. This is implemented by setting the stale bit to 1 on deletion. This reduces the pages needing to be moved during garbage collection

Sequential Updates vs Random Updates:

1) In a SSD, sequential updates would result in the whole block becoming stale. While random updates lead to block becoming partially stale.
2) This makes the garbage collection process more time consuming in case of random updates because the process of garbage collection would now involve reading the correct pages of a block, writing them to a new block, then erasing the initial block
3) In the case of sequential updates, since there won't be blocks which are partially stale we can directly erase the whole block
4) We have observed this by maintaining variables - reads, writes and erases, which are updated whenever a new read or write or erase happens

Sample Output:
Output of a sequence of 20 sequential updates
Before garbage collection: reads - 20 erases - 0 writes - 20
After garbage collection: reads - 20 erases - 20 writes - 20
Output of a sequence of 20 random updates
Before garbage collection: reads - 20 erases - 0 writes - 20
After garbage collection: reads - 35 erases - 32 writes - 35

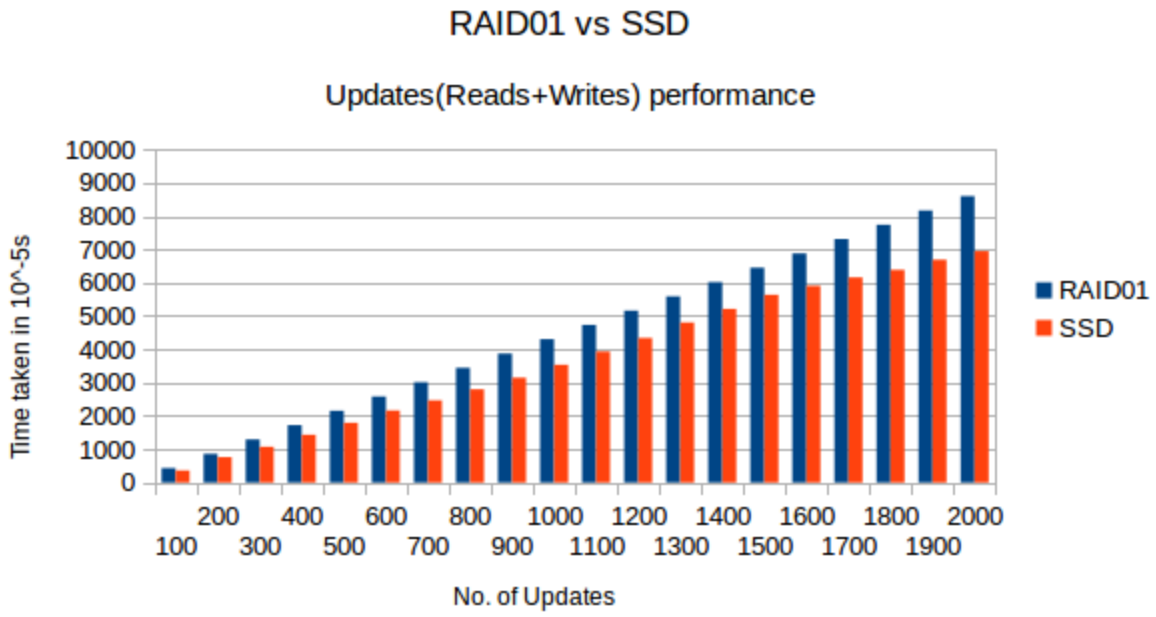
SSD vs RAID1:

1) In RAID1, we have mapped a disk to a file. So, writing to a file would now involve writing to two files.
2) We have used parameters of standard SSD's and HDD's for comparison purposes
   SSD-Samsung 840 Evo 250 GB
   read speed - 510MB/s
   write speed - 377MB/s
   page size - 4KB
   page read - 8us
   page write - 11us
   HDD - WD
   read speed - 201MB/s
   write speed - 180MB/s

page size - 4KB
page read - 20us
page write - 23us

3) In RAID1, a write would comprise of 2 parallel writes to file which in effect is nothing but time taken for a ‘page write’. A read would take time corresponding to a ‘page read’

Data is generated over a file size of 2000 pages, the number of random updates(read + write) or logical IO’s is over the X-axis and on Y-axis is the time taken for this which is calculated by getting the number of physical IO’s and using the parameters mentioned above. It is in units of 10^-5 sec.



References:

1) http://ssd.userbenchmark.com/
2) http://hdd.userbenchmark.com/
3) http://en.wikipedia.org/wiki/Write_amplification
4) http://www.storagereview.com/ssd_vs_hdd