Assignment-3 : Disk Virtualization

Made by Group 08:
Prakhar Kumar 2015CS10667
Gaurav Yadav 2015CS50283
Shreyas Betal 2015CS50293
Shubham Yadav 2015CS50294

3.1 Consolidation & Partitioning

One layer of abstraction is done to see the combination of two arrays as one large array.

Metadata for a disk has:

- Disk Id
- Number of blocks in the disk
- Mapping from Block Id in disk to physical memory address
- List of intervals of blocks allocated to this disk, list has tuples like:
  (length of interval, (start of interval, end of interval))

Unused blocks in the memory are also maintained as intervals, like list of
(length of interval, (start of interval, end of interval)). This list is maintained sorted in decreasing order of length of interval.

When a new disk has to be created, it is allocated blocks from intervals of unused block in decreasing order of length of interval. Blocks are always allocated on one side of the side available interval, not from middle. This reduces fragmentation.

When deleting a disk the intervals allocated to the disk are merged with the intervals of unused blocks.

3.2 Block Replication

Each disk maintains two copies of each block.
Each read has a 10% chance of having error.
Blocks with error are never used again.

Metadata for a disk has:

- Disk Id
- Number of blocks in the disk
- 1st Mapping from Block Id in disk to physical memory address
- 2nd Mapping from Block Id in disk to physical memory address
- 1st List of intervals of blocks allocated to this disk
- 2nd List of intervals of blocks allocated to this disk

When an error is detected, list of intervals of blocks allocated to this disk is modified to remove the error block and allocate a new block to replace it, mapping is also updated. But the error block is not added to list of unused blocks.

The new block which replaces the error block, is allocated from shortest interval of unused blocks, this reduces fragmentation.

3.3 SnapShotting

Metadata for a disk has:

- Disk Id
- Number of blocks in the disk
- List of Mappings from Block Id in disk to physical memory address
- List of, List of intervals of blocks allocated to a checkpoint

We try to save memory by not naively allocating all blocks for each checkpoint. Only blocks which change are allocated.

A list maintains which blocks changed from the last checkpoint.
When a new checkpoint is created, the blocks which did not change from last checkpoint are assigned the same physical memory block as the last checkpoint.

A set of blocks is the current buffer of the disk, all the writes are written into this buffer, when a new checkpoint is created, blocks which changed keep pointing to same physical memory, but the blocks which did not change are pointed to same physical memory as the last checkpoint.

When the disk rolled back to a previous checkpoint A, all the checkpoints after A are removed, if this is not done it will cause branching.