DataMover optimizations

- Anupama Chandwani

Disk types and backing file types

- Types of blocks:
  - Zero block, TBZ, unallocated
- Types of disks:
  - Physical & virtual RDM
  - Thick
    - Eager Zero
    - Lazy Zero (TBZ)
  - Thin/Sparse (unallocated)
- BlockSize of disks:
  - Vmfs snapshot – 512 byte block size
  - Base disk – 1M block size vmfs
  - SeSparse – 8K to 1M block size

Disk types and transformations

- Flat – base
  - Thick (EZT, LZ), RDM
- Sparse – base / snapshot
  - Thin
  - Delta disk / redo logs
- Sesparse – base / snapshot
- Native – VAAI NAS, vVOL - snapshot

Data Mover

- Invoke hardware-based features whenever available, offload copy, xcopy.
- Operations:
  - Zero blocks, clone blocks (fast/full), delete blocks.
- Asynchronous write completion by DM threads
- Submit clone request via IOCTLCMD_VMFS_MOVEDATA in a SG_Array
- DM has a queue of requests & each DM thread asynchronously picks up request from there.
- DM can be used over files, FDS handles, etc.

No hardware offloads

- The VMFS data mover will not leverage hardware offloads -- and will use software data movement instead -- in the following cases:
  - If src and dest VMFS volumes have different block size, fall back to the generic FSDM layer.
  - If src file type is RDM and the dest file type is non-RDM (regular file)
  - If src vmdk type is eagerzeroedthick and dest vmdk type is thin.
  - If either src or dest vmdk is any sort of sparse or hosted format.
  - If logical address and/or transfer length in the requested operation are not aligned to the minimum alignment required by the storage device.
    - VMFS partitions are aligned to 64KB
    - minimum VMFS block size is 1MB
    - most VMFS data movement operations will be 64KB-aligned.
    - However, redo-log files are 512B aligned, so common case for s/w data movement

DM rule book for disk transformation

src	dst	FSSDM – non vmfs DM	FS3DM – vmfs DM
thin	thin	SKIPZERO	Unmapped/tbz src, unmapped dst => zeroes skipped. SKIPFLAG disregarded
thin	ZT	SKIPZERO	SKIPZERO
thin	EZT	SKIPZERO DISABLED	SKIPZERO DISABLED
ZT	thin	SKIPZERO	Unmapped/tbz src, unmapped dst => zeroes skipped. SKIPFLAG disregarded
ZT	ZT	SKIPZERO	SKIPZERO
ZT	EZT	SKIPZERO DISABLED	SKIPZERO DISABLED
EZT	thin	SKIPZERO	As above, SKIPFLAG disregarded. SW DM plugs holes and skips zeroes since dst handle is INVALID.
EZT	ZT	SKIPZERO	Src is mapped, so cannot skip 0 **x**
EZT	EZT	SKIPZERO DISABLED	Src is mapped, so cannot skip 0 **x**

Source: https://wiki.eng.vmware.com/VMawareStorage/InternalFunctionalSpec#VMFS_Data_Movement_Caveats

fs3DM data mover rule book

Src block	Dest block	Action
allocated	allocated	Copy data.
allocated	unallocated	If src block is non-zero, allocate dst block & copy, else skip
allocated	TBZ	Clear TBZ and copy data. (irrespective of src block zero or not)
TBZ/Unallocated	Unallocated	Do nothing
TBZ/Unallocated	allocated	Zero destination for TBZ marked src block
TBZ/Unallocated	TBZ	Do nothing
TBZ/Unallocated	TBZ	Zero destination (SvMotion case: dst is EZT)
RDM	VMDK	FSS DM handles it. Above rules apply
RDM	RDM	Copy everything
VMDK	RDM	Copy data, zero destination for src 0 block

skipZero flag is unset (optimization turned off) when:
- Source disk is a snapshot / delta disk
- Destination is a fully allocated EZT
- This is not a VMFS to VMFS copy.

XvMotion optimizations

- Most optimizations for vmfs 1M block size.
  - vmfs3 mostly 1M blockSize
  - vmfs5 always 1M blockSize
- SkipZero on src: Skip reading and sending zeros.
- SkipZero on dst: Skip writing zeros on destination.
- Prepare dst block based on src (thin only) block type
  - Regular block → batch clear TBZ flag on data blocks. Disable skipZero.
    - Clearing TBZ per IO expensive
    - Since metadata updated, cannot skip writing zeros from src
  - Thin → thin migration = preallocate, punch holes same as src disk
  - Offload batch zeroing to array, if possible.
- Thick → thin migration, cannot plug holes or clear TBZ.

XvMotion optimizations
src == dst == 1M blockSize

Destination format	Optimization
EZT	Batch TBZ clearing for full disk
LZT	Clear TBZ flag for all regular blocks. Skip reading zero blocks from src as dst is already TBZ
Thin	Preallocate regular blocks from src (like fs3dm) Skip reading unallocated src blocks.
Sesparse	Nothing today. TODO for future.

XvMotion optimizations
src != 1M && dst == 1M

Destination format	Optimization
EZT	Skip initial zeroing of disk
Sesparse	SeSparse layer optimizes zeroing IOs

- Dst Thin/LZT
  - Skip writing zeros on dst whenever possible.
  - 1bit/1M bitmap:
    - cloneDroppedZeroBlock
    - CloneFoundContent
  - Read IO, is all 0? (inclined to skip sending 0) but check if content is non-zero, have to send zeros.
    - If this block hasn't been sent, let dest side FS-level zero it. Set droppedZeroBlock
  - Read IO, != 0? (inclined to skip zeroing dest before write) but if droppedZeroBlock set, cannot skip writing zero.
    - If droppedZeroBlock unset, skip writing zeros on dest.