Barry Whyte – IBM Master Inventor – Virtual Storage Performance Architect
June 2013

Optimizing Virtual Storage Performance:

Architecture, Design and Best Practices for Performance

Edge2013

Cloud | Data | Results

June 10 – 14

Mandalay Bay | Las Vegas, NV

Session goals and welcome

- This session will discuss the IBM Storwize family, which includes IBM System Storage SAN Volume Controller (SVC).
- The session will guide attendees to how and why they should configure their disk systems to gain optimal performance for the most common business problems: transactional, analytical and backup performance.
- The session will cover performance at all levels of the disk systems, from basic spindle and SSD performance through to virtual volumes, striping, and advanced functions including IBM's Real-time Compression and Easy Tier functions.
- Attendees will gain a greater understanding of the internals of these systems and be able to put into practice the best configurations in their own or their customers' environments to ensure optimal performance through all layers of the disk systems.

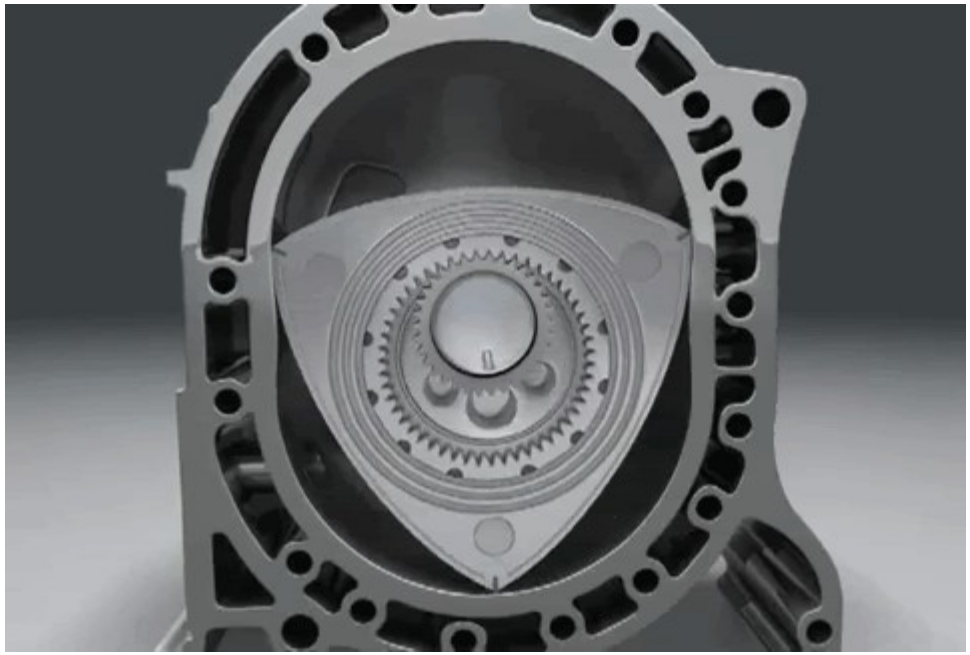
Edge2013
Cloud | Data | Results

Optimizing Virtual Storage Performance

Storage Performance Concepts

Storage Performance - “It depends”

Mazda Renesis rotary engine



Official UK figures – 21mpg urban, 27mpg extra-urban

Achieved figures – 15mpg urban, 21mpg extra-urban

Explanation of official UK government figures

Urban cycle test explained

"It is carried out on a rolling road in a laboratory with an ambient temperature of 20 degrees Celsius (°C) to 30°C. … The maximum speed in the test is 31mph, average speed 12 mph and the distance covered is 2.5 miles"

Extra-urban cycle test explained

"The maximum speed in the test is 75 mph, average speed is 39 mph (63 km/h) and the distance covered is 4.3 miles"

Combined fuel consumption figure

"It is an average of the two tests, weighted by the distances covered in each one"

Your mileage may vary .... or … "it depends"

Taken from :

http://www.direct.gov.uk/en/Motoring/BuyingAndSellingAVehicle/AdviceOnBuyingAndSellingAVehicle/CalculatethefuelconsumptionCO2andtaxcosts/DG_195297

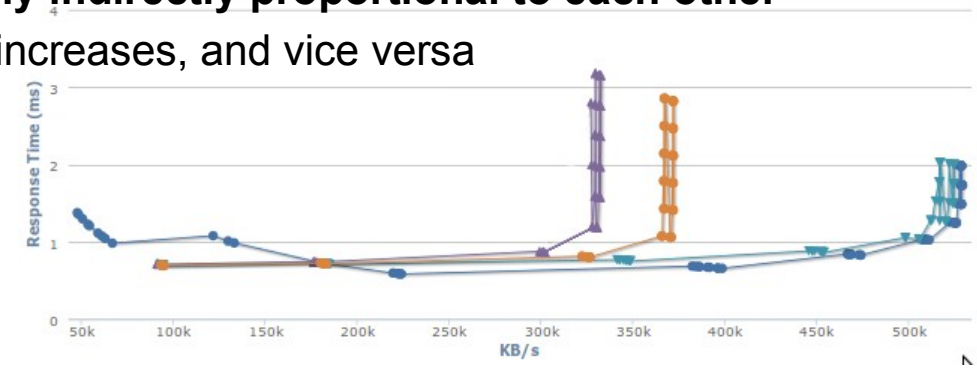
I/O Performance Metrics

- **Throughput**
  - How many Input/Output (I/O) operations per second (IOPS)
  - How many Megabytes or Gigabytes per second (MB/s GB/s)

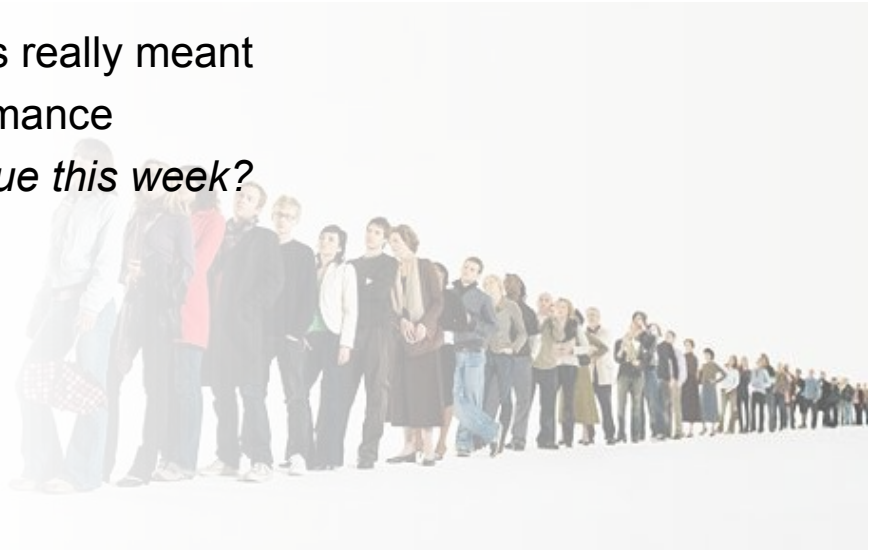
- **Latency / Response Time**
  - How long it takes for an I/O to complete
    - Can be measured from any point in the system;
    - i.e. as seen by the application
    - i.e. as seen by the disk system etc
  - Measured in standard units of time
    - Milliseconds, microseconds or maybe even seconds!

Performance Measurement Criteria

- **Throughput and latency are generally indirectly proportional to each other**
  - As latency decreases, throughput increases, and vice versa
  - Hockey stick curves



- **Queue vs Concurrency**
  - Queue depths often used when concurrency is really meant
  - Queuing can have a dramatic effect on performance
    - *How much time have you wasted in a queue this week?*
  - Host queue depths
  - Storage system queue depths (queue full)
  - Disk queue depths



Optimizing Virtual Storage Performance

SVC and Storwize family architecture

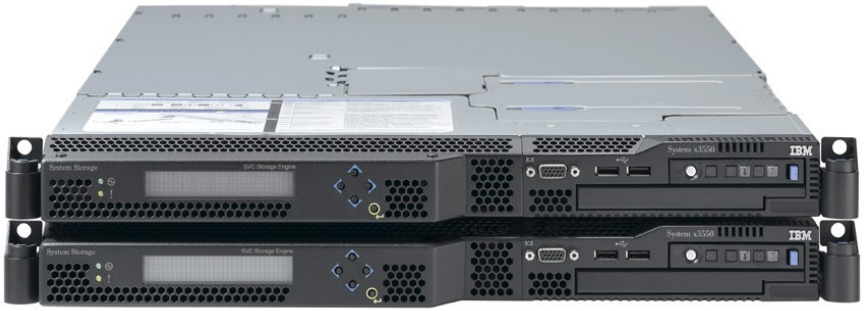
Edge2013
Cloud | Data | Results

SVC and Storwize Family Software Architecture

- **SVC designed around using commodity server hardware**

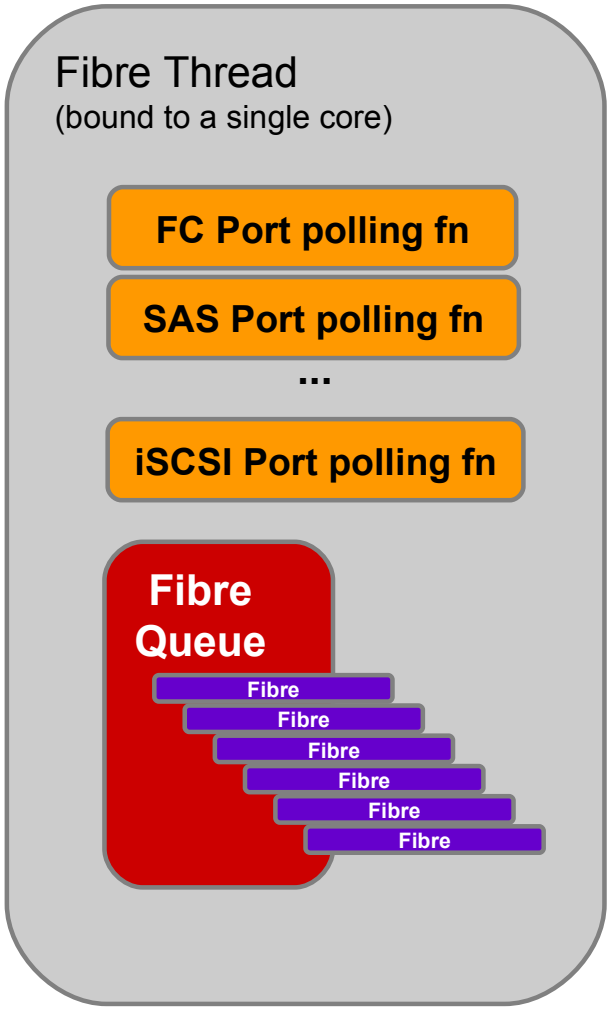
- **In 1999 we realized the future was storage was going to be defined by software**
  - Build a scale out clustered storage platform
  - Utilize standard hardware platforms
  - Add advanced functions in software
  - Divorce the software from the hardware as much as possible
  - First real example of “storage defined storage”

- **Clustering provides scale out for performance and addressability**
  - Design came from IBM Research “Systems Journal, Volume 42, Issue2”
  - Custom memory management
  - I/O process threading
    - Association of cores / ports



SVC and Storwize Family Architecture

- **Since 2007 each node has at least 4 cores**
  - Piggy-back on Intel development
  - Latest and greatest PCIe bus/memory
- **Generally run a single "*fibre thread*" per core**
  - Queues of lightweight fibres
  - No interrupts or kernel code
  - Thread polls hardware from user mode
- **One or more "*driver polling functions*" per thread**
  - Each port has its own polling function
  - Some management functions – like PCIe driver
  - Some kernel functions – like iSCSI
- **Object ownership maybe distributed across cores**
  - Attempt to avoid context switch between threads if possible

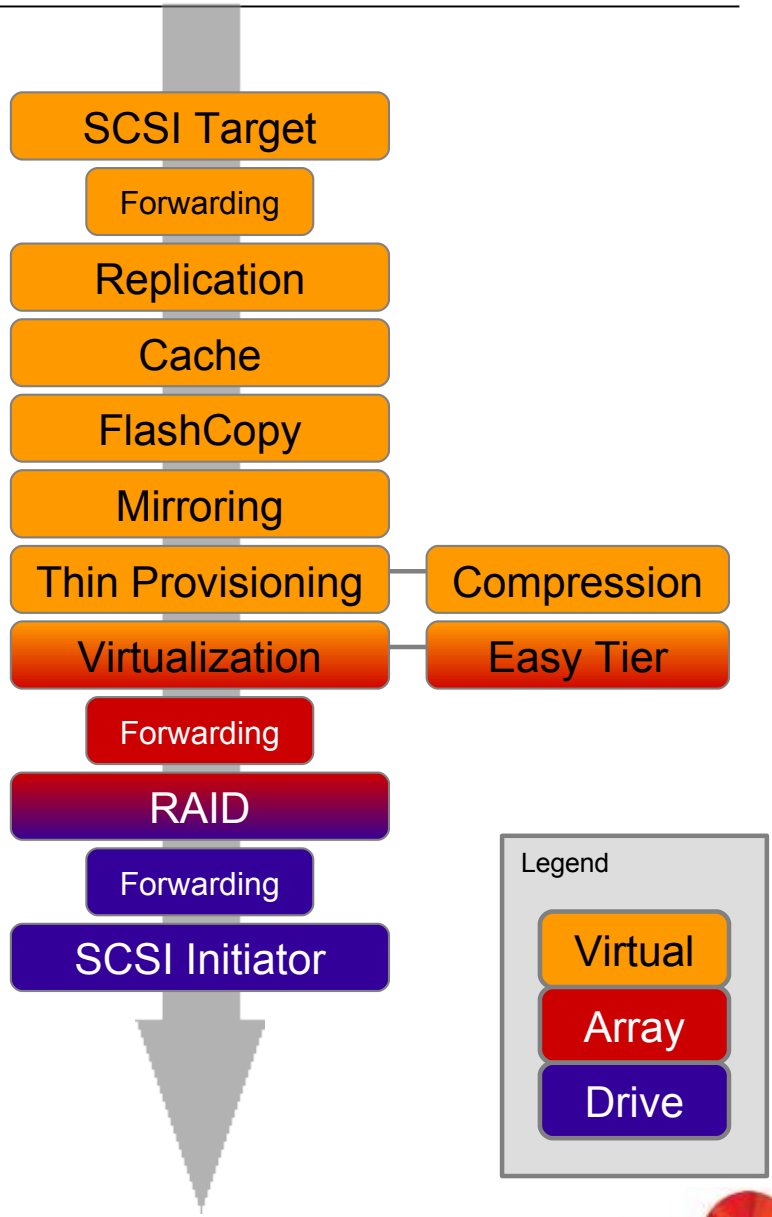


Cores, Ports and Threading

	Fibre Threads	Protocol Ports	Functional Layers
CPU Cores	Fibre thread per core	Driver function assigned to core	Object mgmt assignments per core
Protocol Ports	Driver function per port	Some additional global drivers	
Physical devices	Available to all threads	1 to many assignment	Abstracted by different layers

SVC and Storwize Family Architecture

- **I/O Components form a stack**
  - Host I/O request enters at top, drive or mdisk I/O leave at bottom
- **Interfaces between components is the same**
  - Custom "*platform I/O*" interface
- **Drivers will user mode DMA data from a port**
  - Insert into the SCSI Target layer for the fibre thread associated with the incoming I/O port
- **Forwarding layers allow for I/O to distributed**
  - Passed and processed by another node
  - In case of asymmetric connectivity, path loses etc
- **Orange** components talk in terms of volumes (virtual disks)
- **Red** components talk in terms of managed disks – or arrays
- **Purple** components talk in terms of drives (or on SVC still managed disks)



Optimizing Virtual Storage Performance

Disk and RAID Performance

Edge2013
Cloud | Data | Results

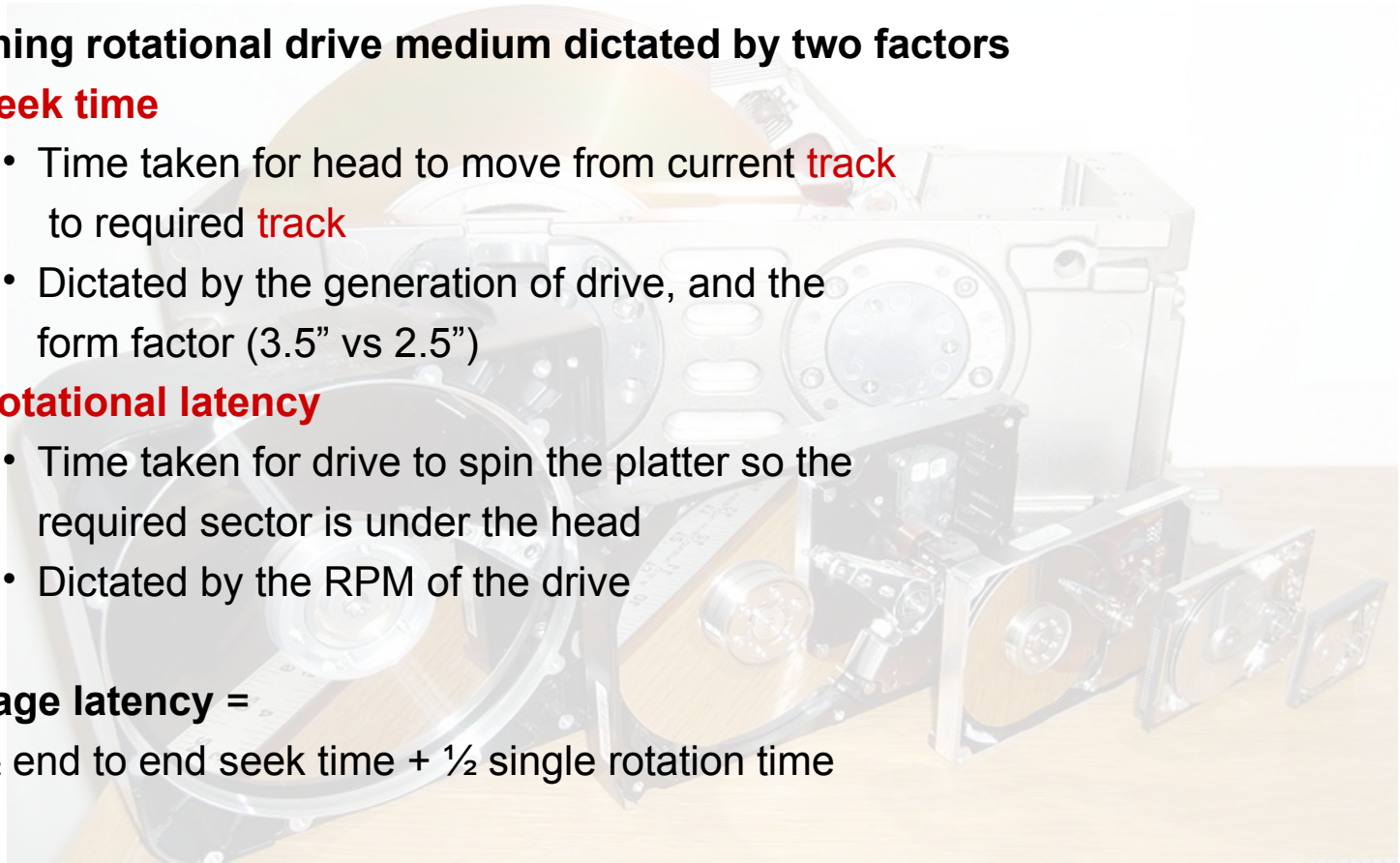
Fundamental Disk Performance - “Spinning Rust”

- **Drive technology really not changed much since original IBM RAMAC drive in 1955**
  - An arm with a read/write head moving across a platter(s) of magnetic sectors.

- **Spinning rotational drive medium dictated by two factors**
  - **Seek time**
    - Time taken for head to move from current track to required track
    - Dictated by the generation of drive, and the form factor (3.5” vs 2.5”)
  - **Rotational latency**
    - Time taken for drive to spin the platter so the required sector is under the head
    - Dictated by the RPM of the drive

- **Average latency =**

  ½ end to end seek time + ½ single rotation time



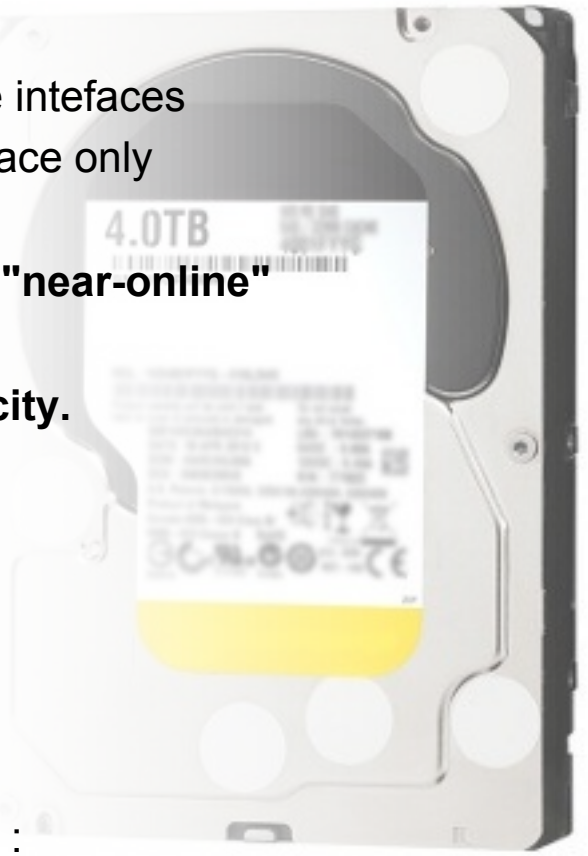
Nearline SAS / SATA - 7,200RPM

- **SAS vs SATA**
  - SAS drives – dual ported – access through two separate intefaces
  - SATA drives – single ported – access through one interface only

- **Nearline as a term was originally used to describe tape, "near-online"**

- **Consumer grade SATA drive technology, but high capacity.**
  - Consumer grade, and highest density
    - = lowest reliability
    - = highest RAID protection needed
  - Generally LFF (3.5”) - currently up to 4TB
  - Some SFF (2.5”) - currently up to 1TB

- **Performance, per 7.2K drive,** we'd normally expect around :
  - **100-150 IOPS**
  - **100-180 MB/s**



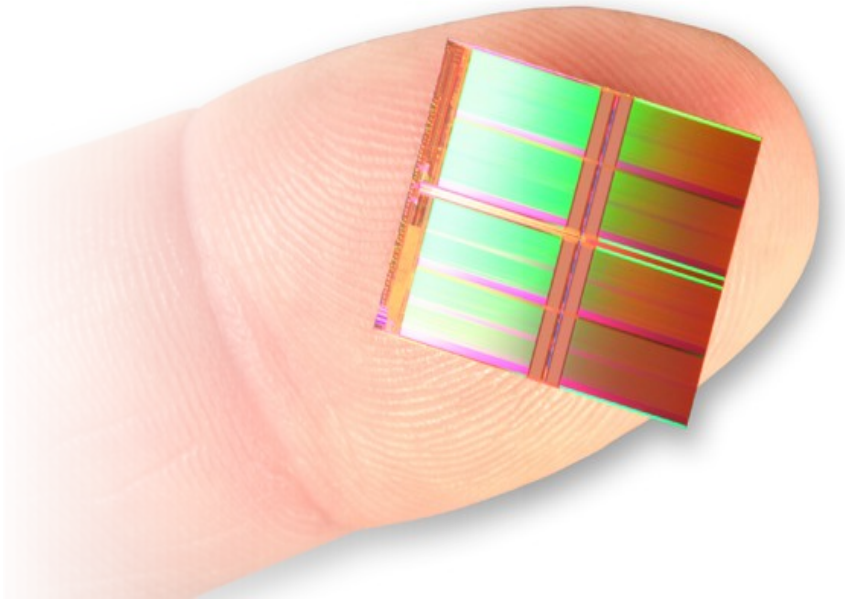
Enterprise SAS – 10,000 / 15,000 RPM

- **Enterprise grade SAS drives, but lower capacity.**
  - Industry has moved to mainly SFF 2.5”
- **Fundamentally different class to NL-SAS / SATA**
  - Not only in terms of RPM
  - Better reliability, firmware, and technology
- **10K RPM**
  - Mid-speed, closer to NL-SAS in capacity (currently 1.2TB)
  - **200-300 IOPS**
  - **120-200 MB/s**
- **15K RPM**
  - Fastest HDD, lowest capacity (currently 300GB)
  - **300-400 IOPS** (some latest generations with short-stroking ~=500 IOPS!)
  - **150-200 MB/s**

Flash drives

- **Flashd drives come in different forms**
  - Consumer
  - Midrange
  - Enterprise

- **The technology can, and does, make a huge difference**
  - cMLC, eMLC, SLC
  - 34nm, 25nm, 20nm...



- **The drive firmware also just as important**
  - Write endurance algorithms
    - Wear-levelling, leveling inactive data
  - Erase before write overheads
    - Pre-emptive reconstruct
    - Pre-erased blocks / over-allocation

Drive Performance Comparison

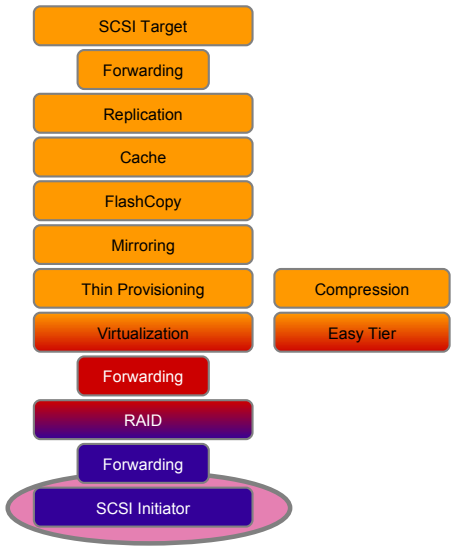
	NL-SAS 7,200 RPM	SAS 10,000 RPM	SAS 15,000 RPM	SSD Consumer	SSD Enterprise
Read IOPS	100 *(150)*	200 *(320)*	300 *(500)*	~5,000	25,000 - 45,000
Write IOPS	100 *(120)*	200 *(275)*	300 *(400)*	~1,000*	7,000 - 30,000*
Read MB/s	100-180	120-200	175-200	200	300-500
Write MB/s	100-180	120-200	150-200	200*	100-500*
Minimum Response Time	9ms	6ms	3.5ms	0.5ms	0.2ms

Values in *(x)* are short stroked (<25% capacity used)
Values in ranges *x-y* vary by drive generation
* Likely to suffer performance drop with sustained writes

(Data recorded from supported drives via Storwize V7000 – single drive, RAID-0, cache disabled)

Storwize Drives - Best Practice

- **SAS networks are a tree network**
  - Cables are trunks – multi-phy cables
  - Enclosures are branches
  - Drives are leaf nodes
- **Single drive failure only loses that leaf (unlike FC-AL etc)**
- **Storwize controllers generally don't have differing performance by enclosure "position"**
  - Logic states that the last enclosure in the chain has the most hops – microsecond latency differences
  - Place SSD in control enclosure, or close to start of chain
- **Each Storwize family member supports different chain lengths, and numbers of chains**
  - Storwize V3700 – one chain, five enclosures
  - Storwize V7000 – two chains, ten enclosures
  - Flex System V7000 – one chain, ten enclosures



RAID Protection Comparison

- **RAID-0**
  - Striping / concatenation
  - No protection against drive loss
  - Capacity = sum(drive capacitites)
- **RAID-1**
  - Mirroring between two drives
  - Protects against one drive loss
  - Capacity = capacity of single drive
- **RAID-10**
  - Mirroring between two sets of striped drives
  - Protects against up to half the mirrored set
  - Capacity = half capacity of all drives
- **RAID-5**
  - Rotating XOR parity
  - Protects against one drive loss
  - Capacity = sum(num drives -1)
- **RAID-6**
  - Rotating double XOR parity
  - Protects against two drive losses
  - Capacity = sum(num drives -2)
- **RAID-X (or distributed)**
  - Generally has one of the other RAID protection types
  - Large number of drives
  - Helps rebuild times

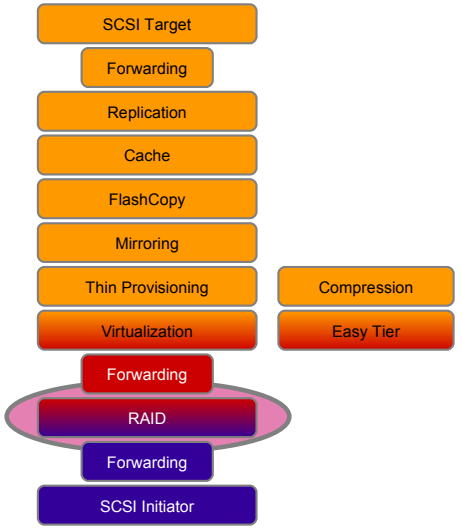
RAID Performance Comparison

	1 Host Read =	1 Host Write =
RAID-0	1 disk read	1 disk writes
RAID-1 or 10	1 disk read	2 disk writes
RAID-5	1 disk read	2 disk reads and 2 disk writes
RAID-6	1 disk read	3 read reads and 3 disk writes

- **Random Performance is dictated by the overheads on writes**
- **RAID-5 and 6 will give best capacity usage, and failure protection**
  - Write penalty is 4x for RAID-5 or 6x for RAID-6
  - So you need to consider this when creating your arrays.
- **NL-SAS being "consumer" grade and most liable to failure**
  - Typically needs **RAID-6**
  - **Catch 22** – worst performing drive, needs worst overhead RAID!

Storwize RAID Best Practice

- **As discussed, ports and objects are assigned to cores (fibre threads)**
- **In this case, Storwize RAID array objects are assigned to a thread**
  - All caching of RAID "strips" common to one thread
  - All XOR and manipulation of array "stripes" common to one thread
  - Reduces contention and lock requirements in code – thus faster!
- **For a given system, optimal core usage means to create at least as many arrays as you have cores**
  - Since arrays can be accessed via both nodes in an IO Group, only the number of cores per node is important
  - Less than 4 arrays generally only a limit if using SSD arrays
- Each node in the Storwize family has different core counts:
  - SVC CG8* – 6+ cores – but 4 fibre threads
    - Older SVC nodes 4 cores (8G4, CF8)
  - Storwize V7000 and FlexV7000 – 4 cores
  - Storwize V3700 – 2 cores

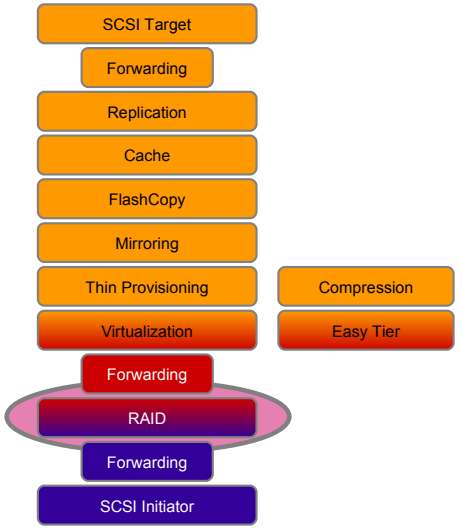


* Some early CG8 nodes have 4 cores

Storwize RAID Best Practice – strips and stripes

- **A RAID array has N component drives**
  - The logical block address of the array is striped across all drives
  - Each drive presents a “strip” of capacity, before moving on to next drive
- **A full stripe spans one strip from each component drive**.
  - sometimes called full stride

- **Storwize controllers support 128KB and 256KB strips**
  - Configurable on CLI at array create time
  - 256KB default
- **Random write workloads**
  - Equal chance of hitting any strip, either size fine
- **Sequential write workload considerations**
  - Host I/O large enough for a full stripe?
  - i.e. 1MB = 8x128KB, 2MB = 8x256KB etc
    - So we can build parity in memory with no drive reads
  - R5 “Full stripe write” = ~1/3 the disk IOPs and MB/s
  - R6 “Full stripe write” = ~1/5 the disk IOPs and MB/s



Edge2013
Cloud | Data | Results

Optimizing Virtual Storage Performance

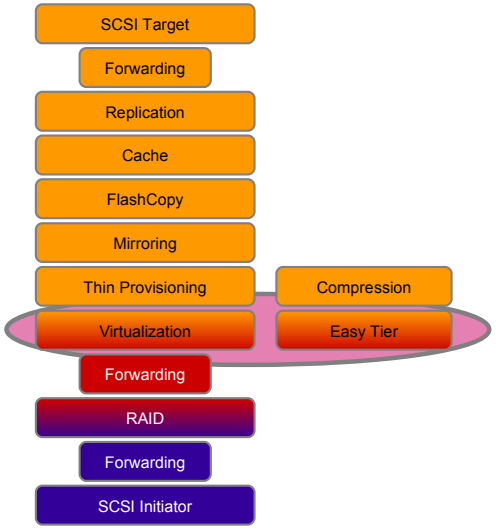
Storage Pools and Volumes

SVC and Storwize – Storage Pools Best Practice

- **A storage pool contains one or more arrays (or managed disks on SVC)**
  - The storage pool maps between a virtual disk and managed disks (mdisk)

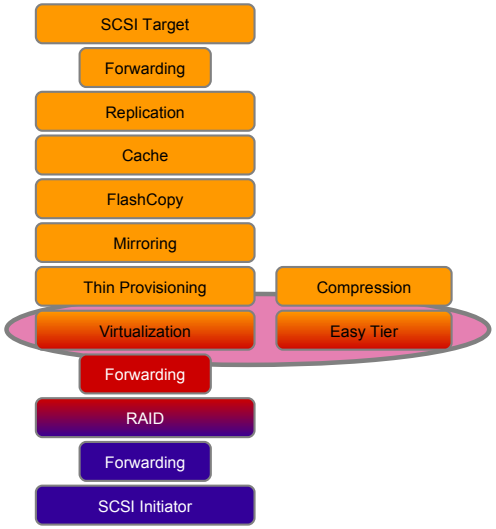
- **Storage pools have an "*extent size*" 16MB up to 8GB**
  - Default extent size before 7.1.0 code is 256MB, 7.1.0 + it is now 1GB
  - Storage pools are like RAID-0 arrays
  - Extent is like a RAID strip
  - Recommended to use same extent size for all pools in a given cluster – where possible
    - Enables simplified volume migration

- **Should contain only one class of array / mdisk**
  - Same reliability level (RAID)
  - Same drive speed
  - Also for SVC usually the same virtualized controller model
  - Exception being "*hybrid pools*" - when using Easy Tier



SVC and Storwize – Storage Pools Best Practice

- **Storage pools used to coalesce 'like-minded' capacity into single manageable resource**
  - Simple tiering construct – pool by tier, and by reliability
- **Wide striping enables greater performance for a single volume**
  - **Random I/O** workloads benefit from many more drives contributing their IOPS
- **Recommendations for sequential I/O processing**
  - Sequential read/write workloads generally dictated by single drive/array performance
  - Low queue depths, and hitting consecutive LBAs
    - Some concurrency at RAID level
    - Extent striping has higher granularity
  - In certain cases, where a workload is always MB/s orientated, single mdisk and volume per pool can provide best MB/s
  - Alternatively use sequential volume options.

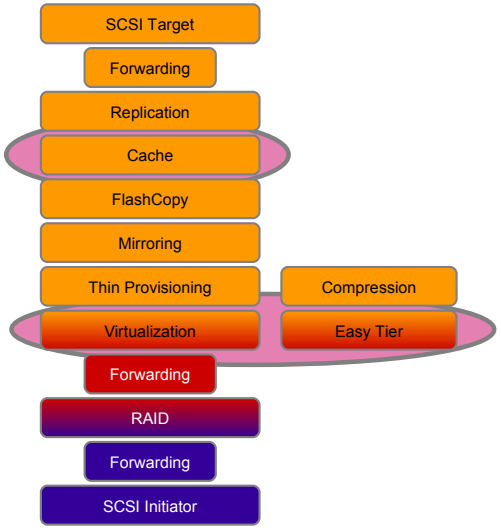


SVC and Storwize – Storage Pools – Cache Partitioning

- **SVC and Storwize family cache include an automatic partitioning scheme**
  - No user configuration needed

# Storage Pools	Upper Partition Limit
1	100%
2	66%
3	40%
4	30%
5+	25%

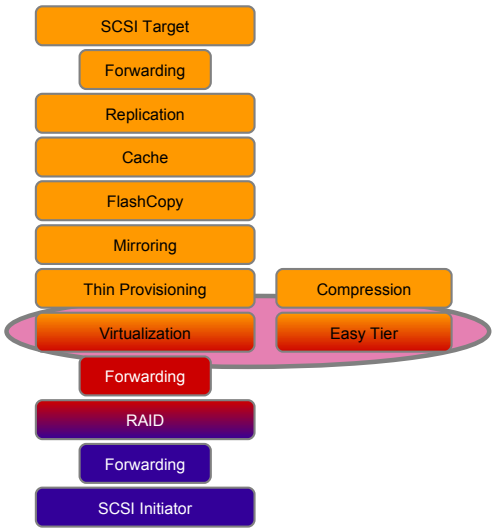
- **Cache partitioning originally introduced to SVC to prevent a single overloaded virtualized storage controller consuming 100% of write cache**
  - Each Storage pool results in the creation of its own partition
  - Each partition has a "% of write cache use limit"
  - Stops cache contamination of workload across pools
  - Only write I/O are partitioned
  - Don't think of this as a limit or causing a problem
  - If a partition reaches its limit, then you have an overloaded Storage pool – wrt writes



- **SVC considerations**
  - Storage pool per controller / tier of storage
  - Storwize system behind SVC
    - On Storwize system create a single per array and stripe at SVC

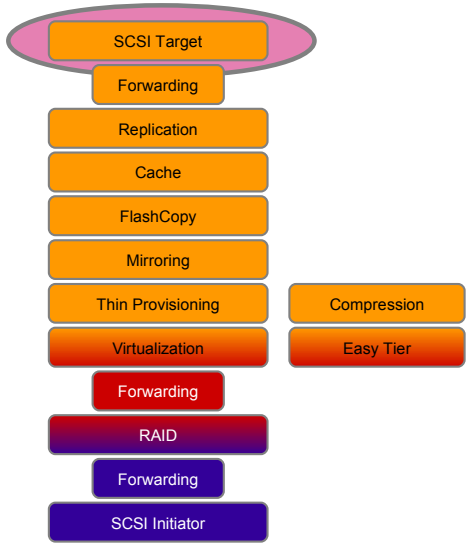
SVC and Storwize – Hybrid-pools – Easy Tier

- **Easy Tier pools are the exception to the "single class per pool" rule**
  - Each mdisk/array in a pool is assigned a technology type
    - Internal storage arrays inherit tech type from drives
    - External storage mdisks – user needs to define tech type
- **Storage pool extent size, defines granularity of movement**
- **Typically expect less than 5% of SSD tier needed in pool**
  - Will always try to use available SSD
- **Analysis tool can determine performance benefit**
  - In terms of capacity of hot data, and expect performance
  - Can be run on a system without hybrid pools
- **Easy Tier – at what layer? SVC or underlying controller**
  - SVC overarching enterprise view
  - Internal SAS vs external FC migration bandwidth



SVC and Storwize – Volumes

- **Volumes are created from capacity in a single Storage Pool**
  - Hence volumes inherit performance and reliability of the pool
- **Striped or sequential**
  - Striped volumes concatenate one extent from each mdisk in turn
    - Random starting mdisk
    - Best for transactional, random performance applications
    - Single vdisk *could* provide sum of pool's performance
  - Sequential volumes allocate extents sequentially from a single mdisk
    - Starts from first free extent on given mdisk
    - Best for sequential, streaming, backup
    - Single vdisk limited to single mdisk performance
    - Sequential performance usually dictated by single array anyway (consecutive blocks)
- **Thin provisioned**
  - Performance should be similar to fully allocated
  - Watch out for fast growing volumes

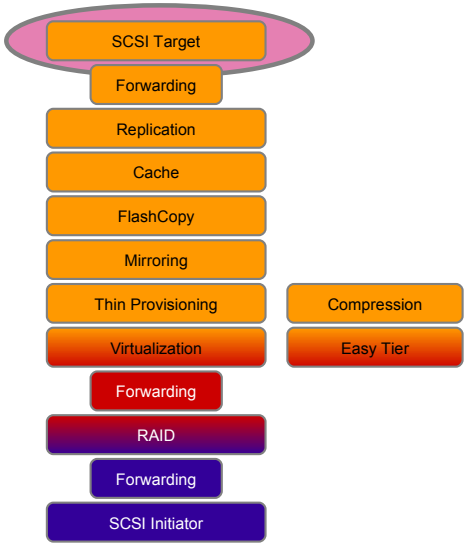


SVC and Storwize – Volumes and Caching

- **Cache coalescing / spliting of volume I/O**
  - Cache will send at most 256KB read I/O down the stack
  - SVC cache will send at most 256KB write I/O down the stack
  - Storwize cache will send up to a full stripe write I/O down the stack
    - All systems will attempt to coalesce smaller sequential I/O into these sizes
  - Exceptions: where a component below the cache requests smaller I/O
    - Grain size – FlashCopy, Volume Mirroring, Thin Provisioning, Compression

- **Preferred nodes / ALUA / Active Active**
  - SVC and Storwize are A/A with a preference
  - No overhead for ignoring preference
  - Preference defines which node owns the volume
    - Owner node will destage for that volume
    - Better read hit chance if preference followed

- **Queue depths, concurrency and volume counts**
  - Virtualization is still bound by the laws of physics!



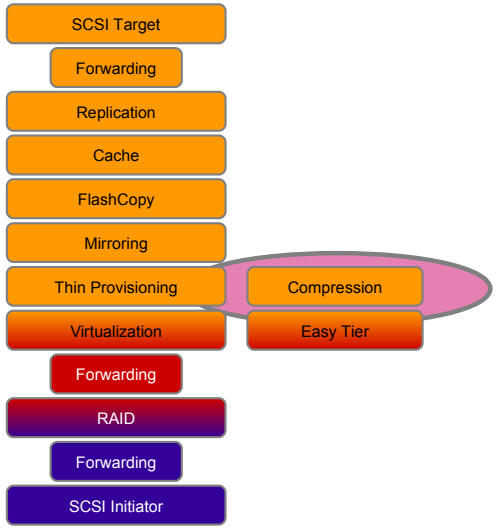
Edge2013
Cloud | Data | Results

Optimizing Virtual Storage Performance

Advanced Functions

SVC and Storwize – Compression

- **Compression could be a session in its own right**
  - Use compresstimator!
- **Temporal locality – no time machine yet... working on it**
  - The predictability of a workload, pseudo random
  - Is the data read back in the same pseudo random order it was written
  - Is there a time locality to the data pattern
- **Transactional, Virtual Machine, and small block work well**
  - Sequential streaming, may become mips limited
- **Implications on cores and existing workloads**
  - Enabling compression may dedicate cores
  - See 'Best Practice' Redbook



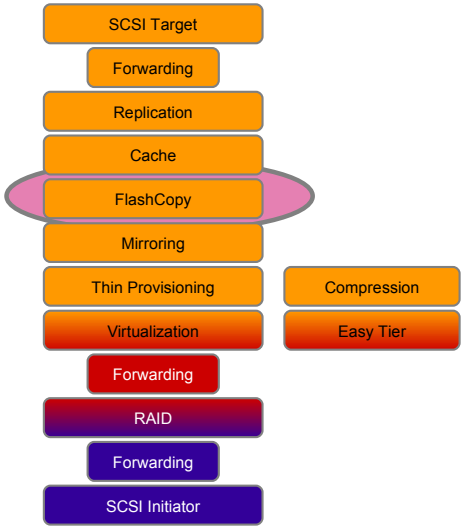
SVC and Storwize – FlashCopy, Volume Mirroring, Replication

- **These functions all create additional mdisk I/O**
  - FlashCopy – read and two writes – for a single volume write
  - Mirroring – two writes – for a single volume write (but written to two different pools)
  - Replication – two writes – for a single volume write (but written to two clusters)

- **FlashCopy**
  - Prepare for consistent copy requires cache flush
  - Very heavy write workloads can become target limited
  - Dependency chains – beware of overloading copies

- **Volume Mirroring**
  - Balance primary copy – to balance read load
  - Copies should be to comparable mdisks - writes

- **Replication**
  - Secondary, secondary, secondary...
  - GlobalMirror RPO – very short
  - Global Mirror with change volumes – prepare



Summary

- **Ultimate drive performance in a given system dictates performance capability**
- **RAID level will impact write performance**
- **Storage Pools help consolidate and boost single volume performance**
- **Volumes will inherit from Storage Pool performance**
  - Advanced functions may alter Volume performance
- **As with all things performance, "it probably depends..."**