VA 7000 Series

VA7000 Series Storage Virtualization and AutoRAID

Roger Buckthal

level400 training

firmware

agenda

- product overview
- management demo
- background
  - VA array objectives
  - fundamentals principles
- firmware architecture
  - IO flow through the firmware
- RAID Levels
  - RAID 1+0
  - RAID 5DP
- data layout
  - usable capacity computation
- AutoRAID policies

level400 training

firmware

Quick View Stats

Shared

- 3U Rack Mount Enclosure w/15 disks
- Controller and 15 disks in single shelve
- Optimized for 1" mechanism form factor
- 18GB, 36GB and 73GB disks, 10 and 15K RPM
- Redundant and hot-plug PS, fans, controller, ….
- Up to 2G Bytes (total) mirrored write cache

VA7100 (Cassini)

- 3 to 15 disks – single enclosure
- rackmount or deskside
- 1Gb/s host and disk FC
- 90MB/s, 12K IOPs cache, 3K disk (reads)

VA7400 (Cronus)

- 8 to 105 disks - 6 add-on shelves
- rackmount only
- 2Gb/s host, 1Gb/s disk (upgradeable w/VA7405
- 160MB/s, 30K IOPs cache, 9K disk (reads)

level400 training

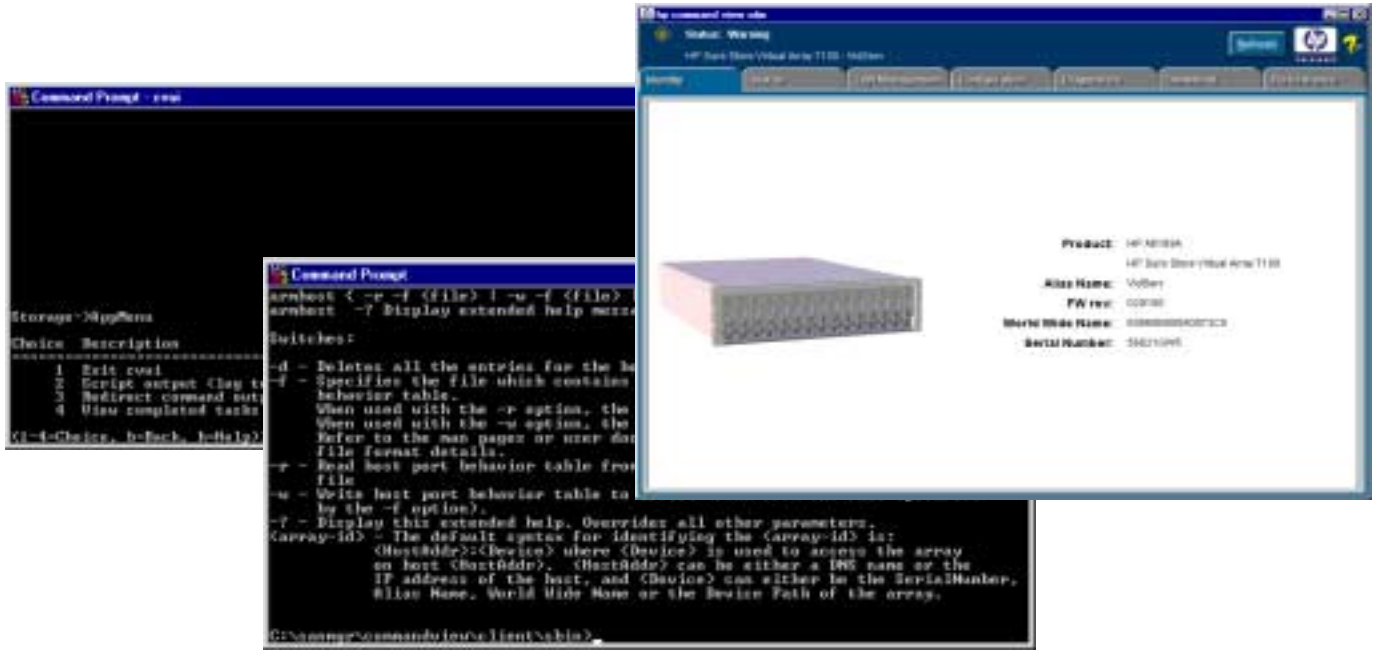
firmware

Productoverview

level400 training

firmware

command view sdm



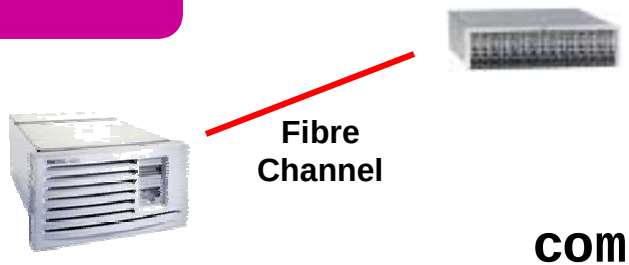
Storage manager for the VA7000 series

level400 training

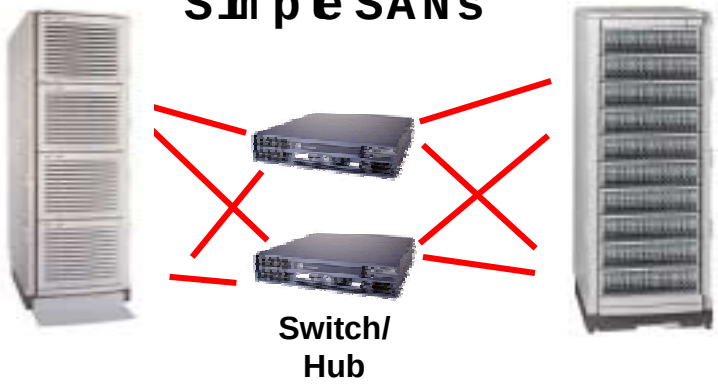
firmware

storage management solutions for all environments

Direct Connect

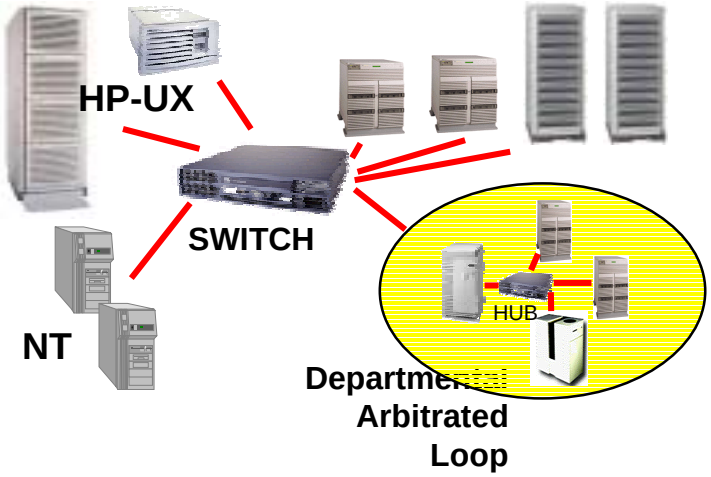


Simple SANs



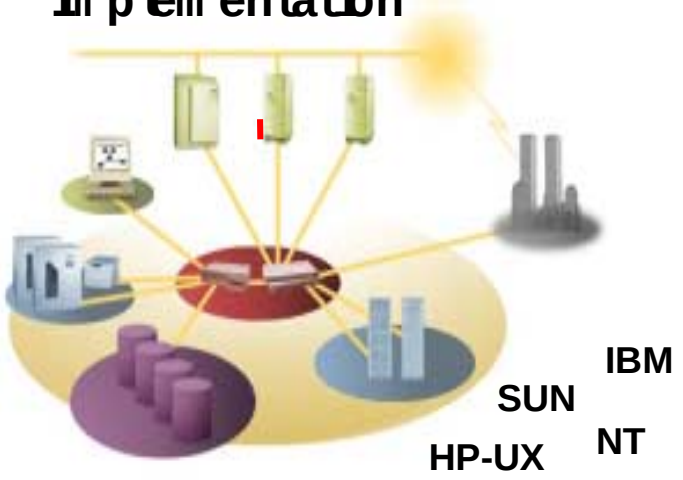
command view sdm

Heterogeneous Clusters



OpenView Storage Node Manager

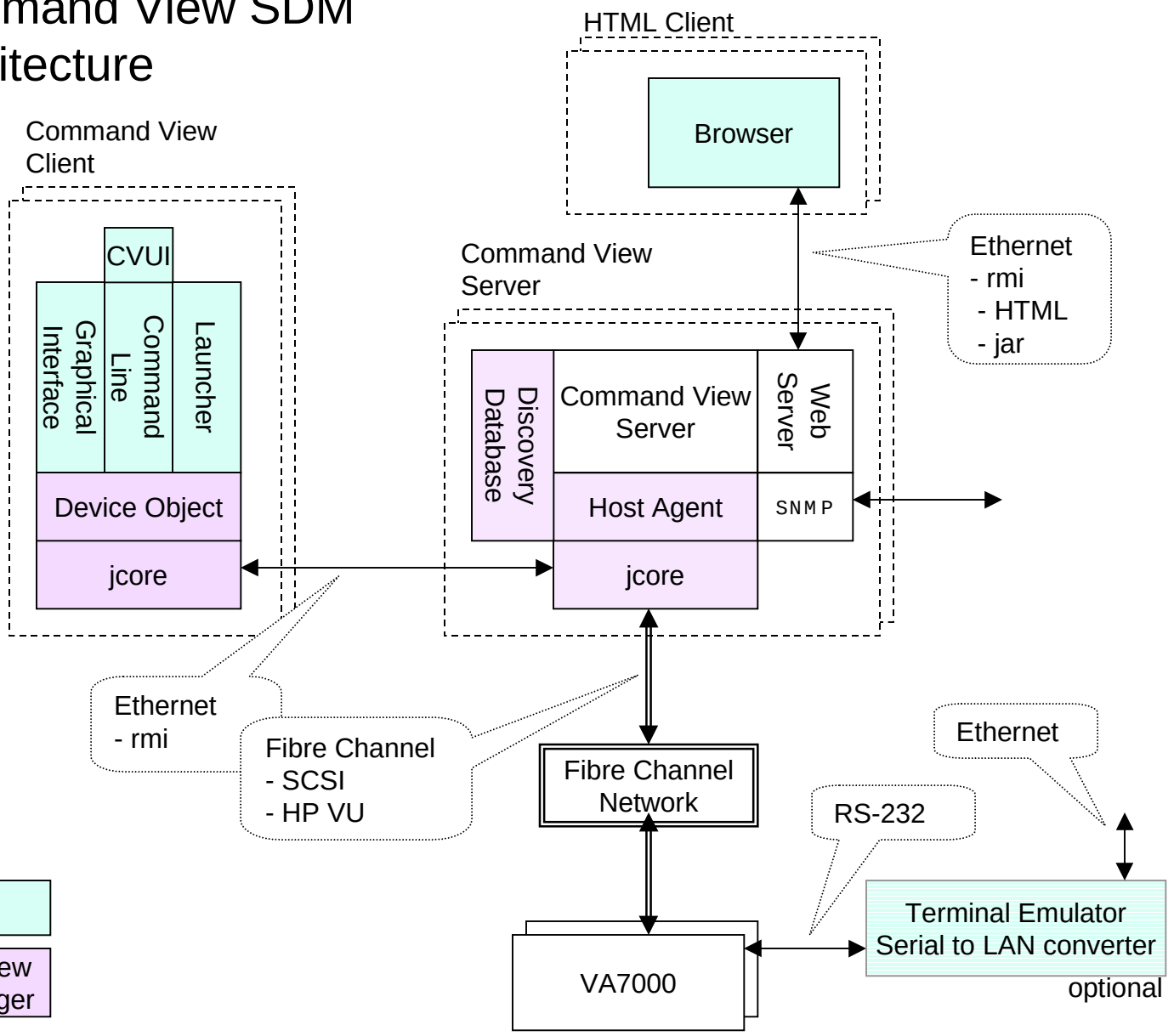
Enterprise-wide SAN Implementation



level400 training

firmware

Command View SDM Architecture



level400 training

firmware

caveats and acknowledgements

- this is a simplified description for the first release for the VA7400 and the associated firmware upgrade for the VA7100 – some tuning is still in progress, thus this presentation may not represent the released product.
- this release is not final, or perfect. we know of many possible improvements to the algorithms.
- as you learn the internal algorithms of the array, it will be easy to hypothesize workbads that can defeat the goals of the algorithms. we are interested in specific, actionable feedback to improve the system – please feel free contact me after this session.
- many of our customers (and others) have a valid concern about the automation designed into this array. we acknowledge that and are working on implemented traditional array configuration for the va series.

level400 training

firmware

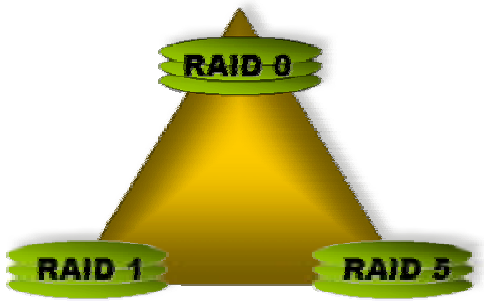
virtual array objectives

- lower management costs. storage capacities are increasing at a faster rate than skilled storage administrators. products that allow significantly fewer storage administers will have a competitive advantage.
- provide superior price/performance. scale the quantity of arrays to meet any absolute performance (and capacity) requirements – a network modular arrays
- provide superior availability and software features for mid-range arrays - match or bet the competition

level400 training

firmware

fundamentalprinciples

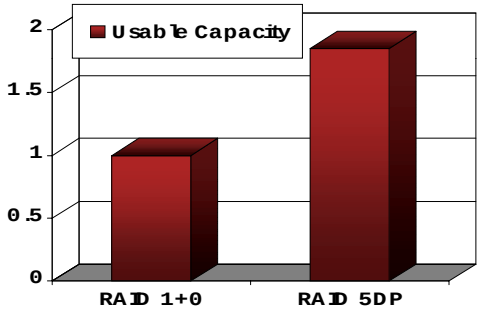


Performance - Cost - Availability

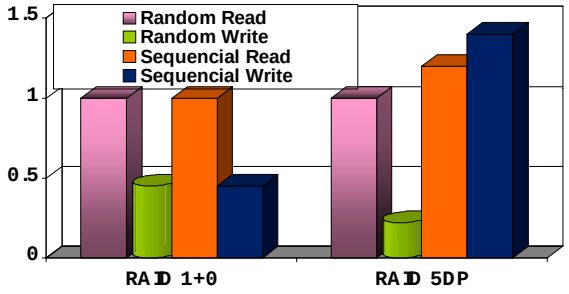
the storage efficiency, performance and availability **characteristic vary** by RAID level. a dynamic system, like a cache hierarchy, can provide performance characteristics similar to the high performance component of the hierarch at a cost similar to the lowest cost component of the hierarchy.

locality of access - recently accessed data is most likely to be access in the future

- RAID 1+0 : *best for OLTP/general file systems*
  - superior performance for small (<64K) random write workbads
  - most redundancy overhead
- RAID 5DP : *best for OLAP and rich media*
  - highest availability
  - least redundancy overhead (>5 disks)
  - superior (theoretical) sequential read and write performance
  - lowest random write performance
- Either RAID level
  - equivalent performance for random read workbads



Capacity

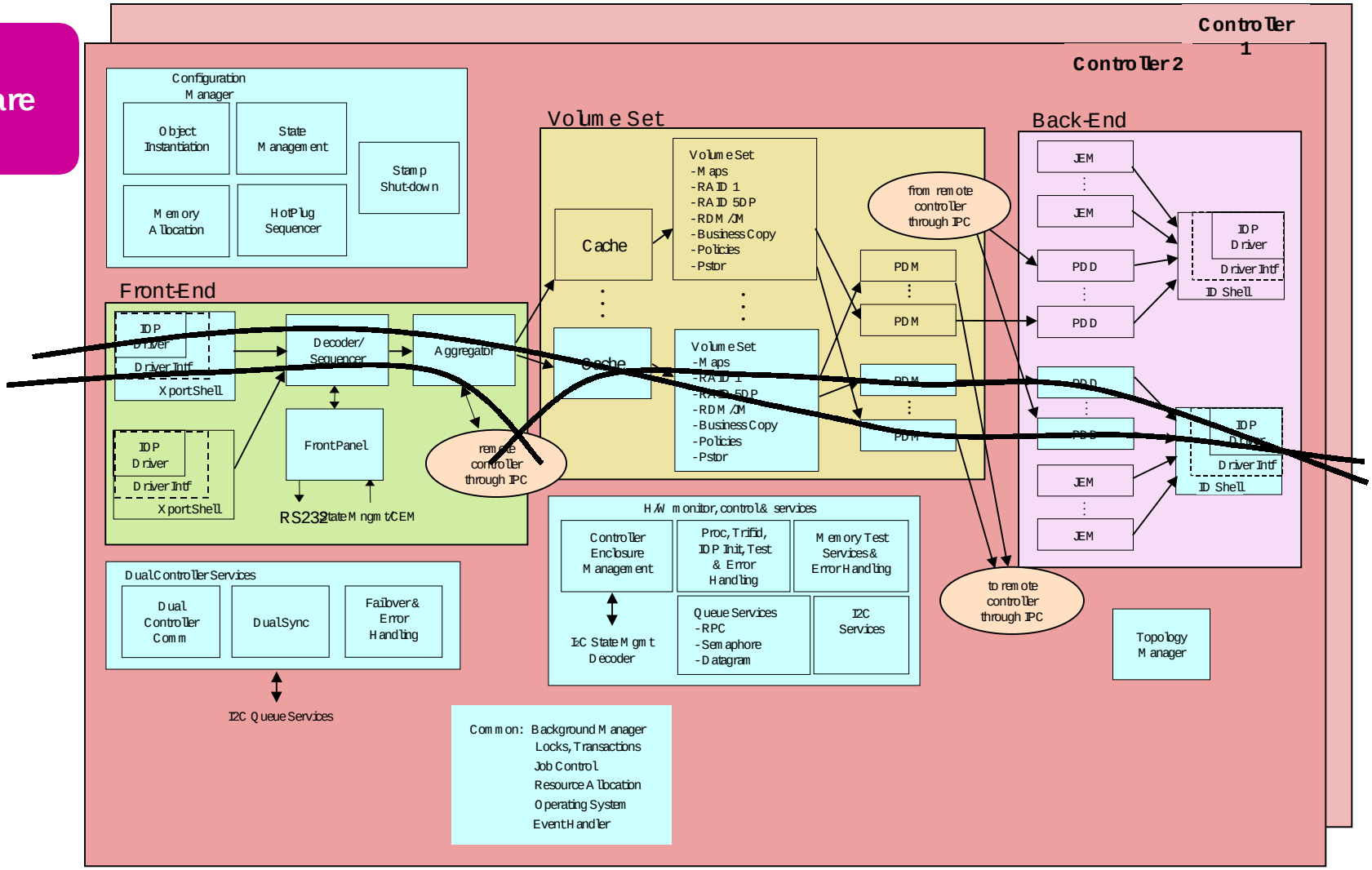


Performance

level400 training

firmware

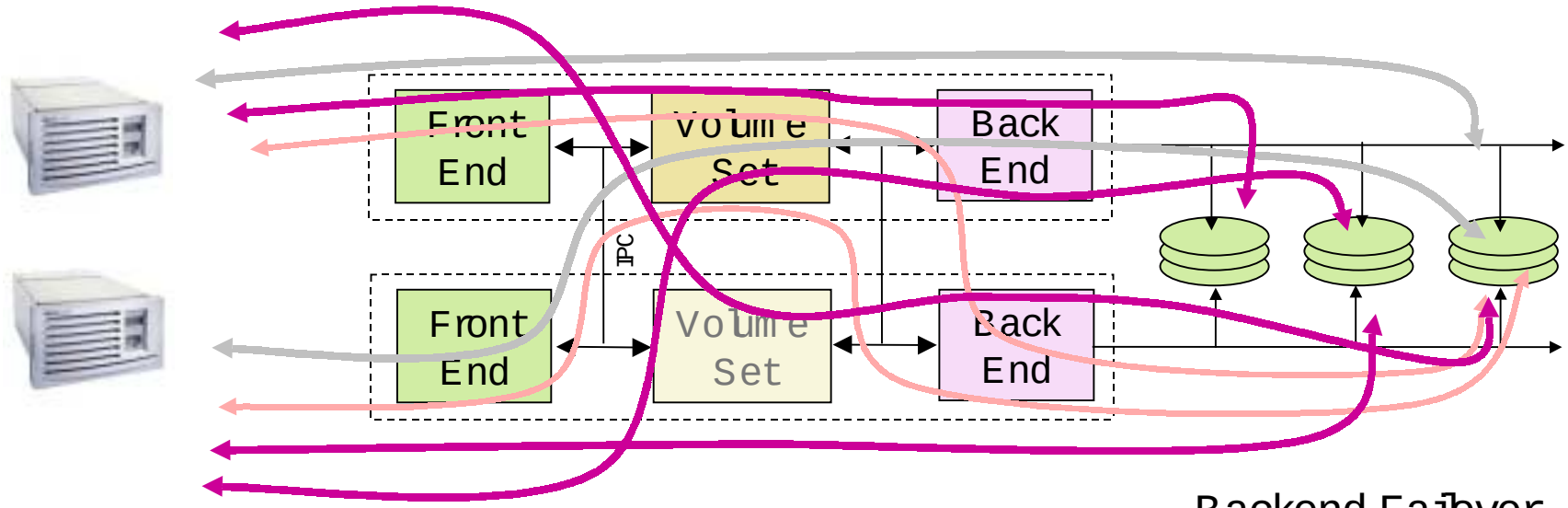
firmware architecture block diagram



level400 training

firmware

VA7000 IO Flow



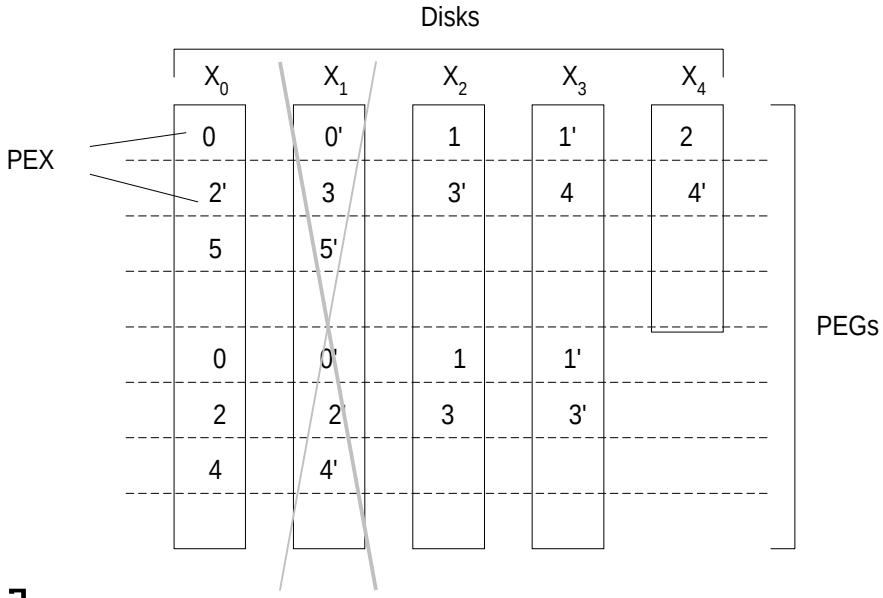
7100 Normal

7100 Back End Error

VA7400 Normal

Backend Failover
- disk by disk
- assign on up

RAID 1+0



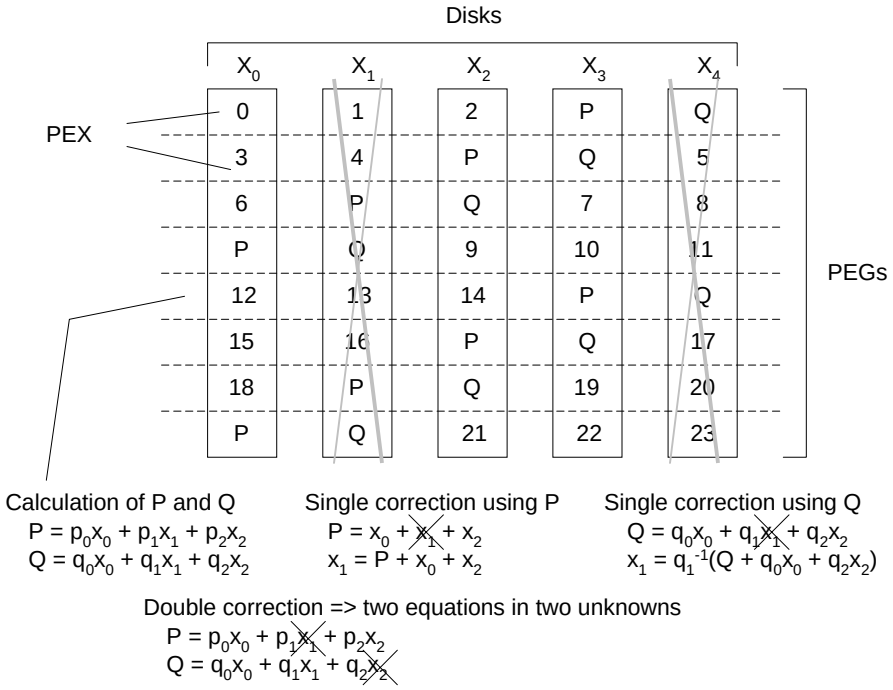
- striped and mirrored
- tolerant to some dual failures
  - even number of disks in PEG
    - left or right (just one) logical disk critical
  - odd number of disks in PEG
    - both left and right logical disk critical
- disk mapping (left or right) not visible to outside array

level400 training

firmware

RAID 5DP

***White paper available on ESP on RAID DP and data protection**

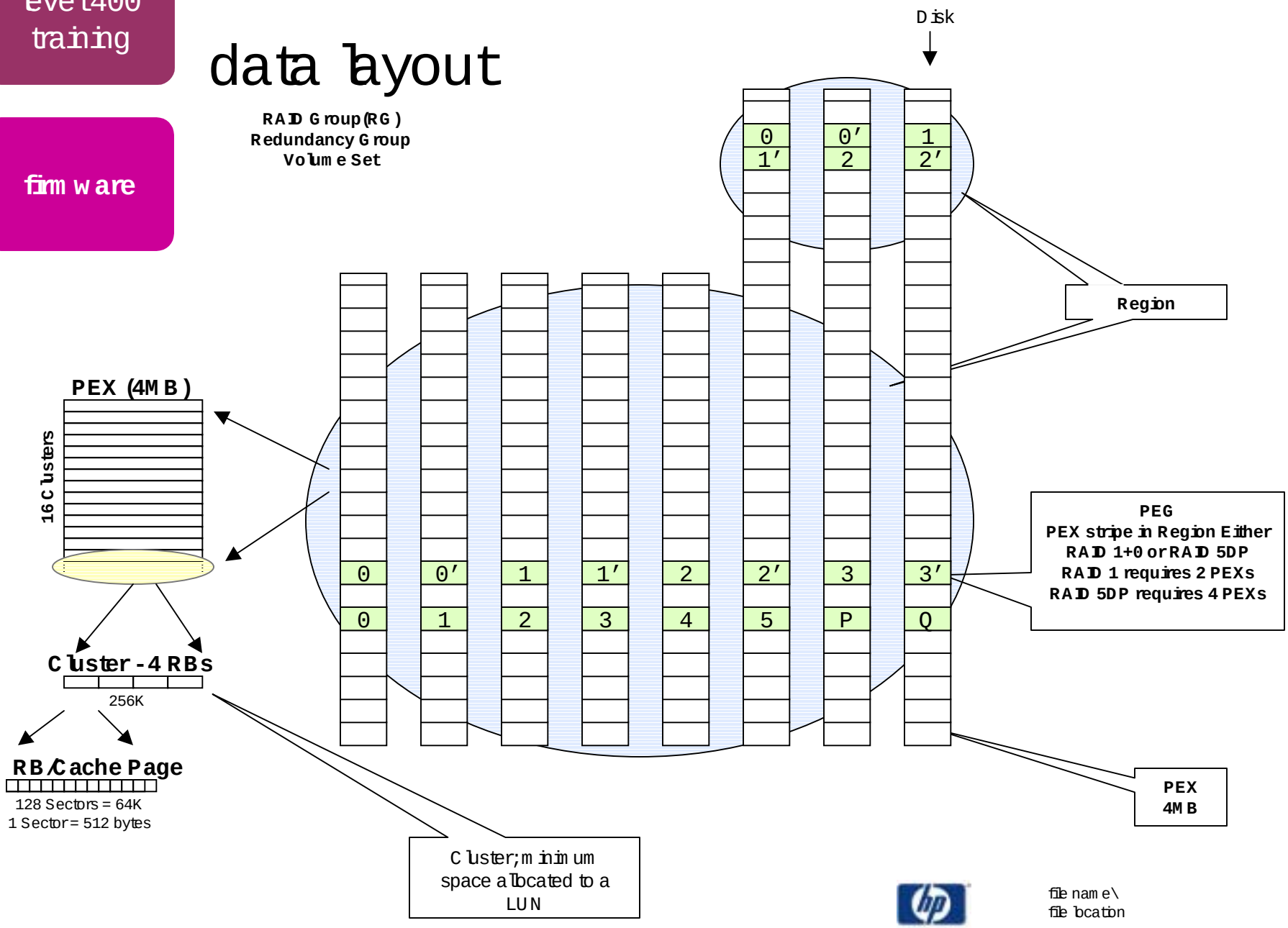


- N+2 rotating error correction - similar to RAID 5
- P is XOR (parity)
- Q is reed-solomon ECC coding
  - similar to RAM and disk error correction codes
  - Q = 'linearly independent' of P
- tolerant to double failures – two equations in two unknowns
- all in hardware – as fast as XOR

level400 training

firmware

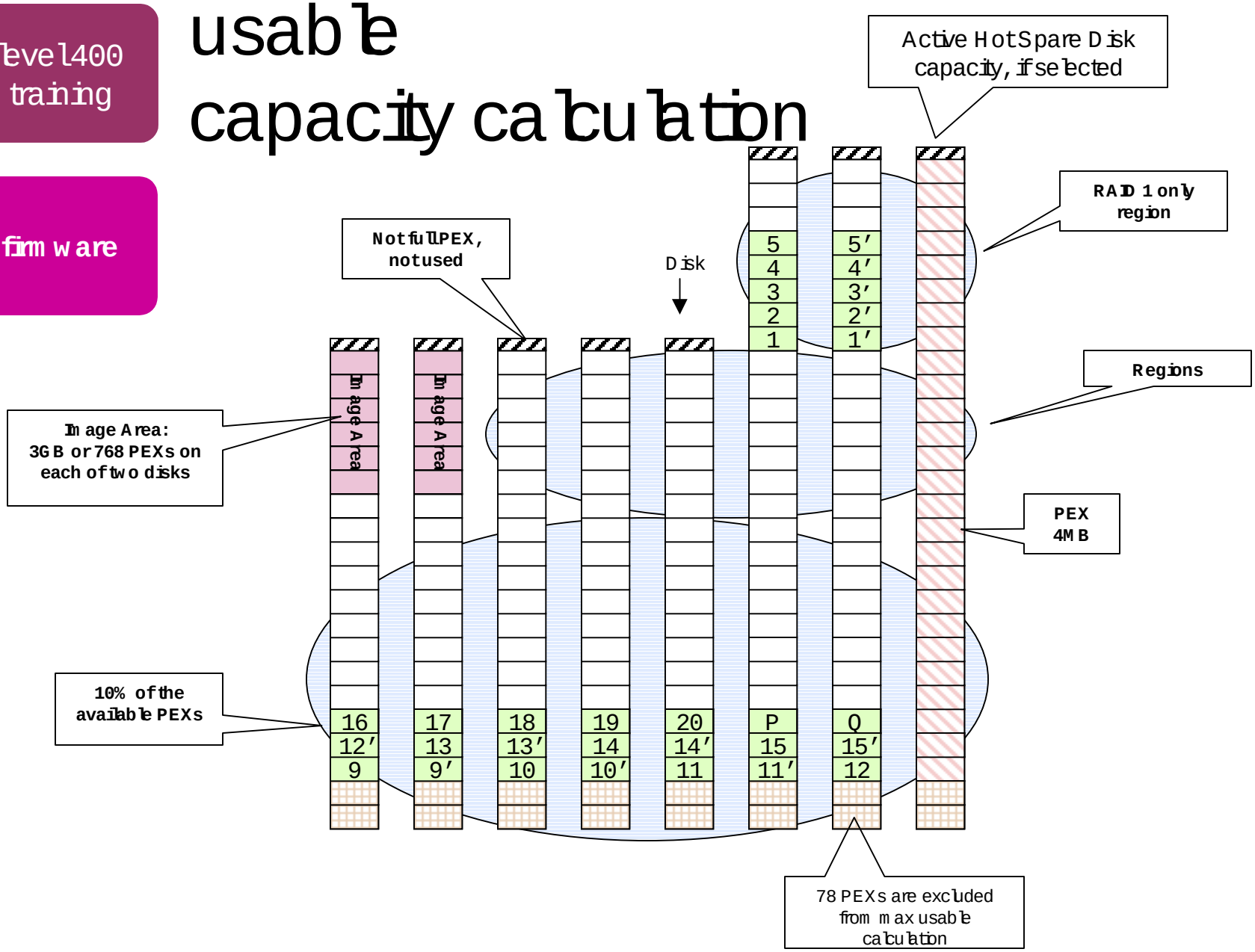
data layout



level400 training

firmware

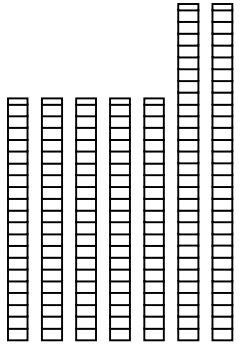
usable capacity calculation



level400 training

firmware

the 'real' data layout



- it's dynamic - based on workload and rules (policies) of the game
- VA7100 has 1 Redundancy Group (RG), VA7400 has 2 RGs
- disks are assigned to RG by initial enclosure slot location - odd/even
  - once assigned to RG, disk can be moved to new slot, but will not change RG
  - more odd slots than even slots
- LUNs will be assign to a RG at LUN creation time
  - each RG is like a separate array

level400 training

firmware

the basic game

Sector – 512 bytes
RB/Cache Page – 128 sectors, 64K
Cluster –4 RBs, 256K
PEX -16 Clusters, 4MB
PEG – stripe of PEX's

- LUN creation reserves space from available clusters
- 'new writes' (space not previously allocated) cause allocation of physical space (cluster)
- array fills with new writes
- reads do not cause data movement – read in place
- re-writes always occur in place (to the same cluster/ RAID level)
- policies move clusters between RAID levels or disks

performance - availability - cost

level400 training

firmware

the policies

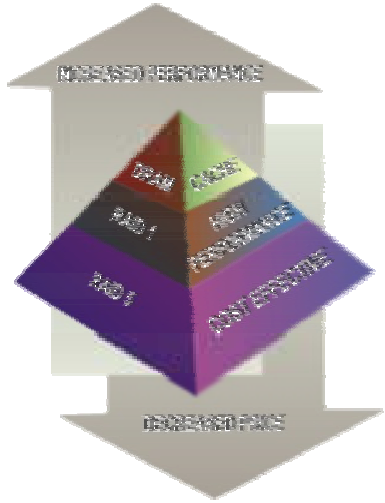
- data placement – *the rules on where data is written*
- makespace – *creates more free-space to allow a new-write to complete* - foreground conversion of clusters from R1 to R5DP
- optimize – *creates more free-space for future new-writes or promotions by moving inactive write data to R5* - background conversion of clusters from R1 to R5DP
- promote – *gets clusters that are recently written randomly to R1* - background conversion of data from R5DP to R1
- balance – *re-distributes data after new disks are added* - extend PEGs
- rebuild – *removes failed disk from RG* - shrink PEGs
- scrub – *tests and corrects latent un-recoverable disk errors* - read every sector

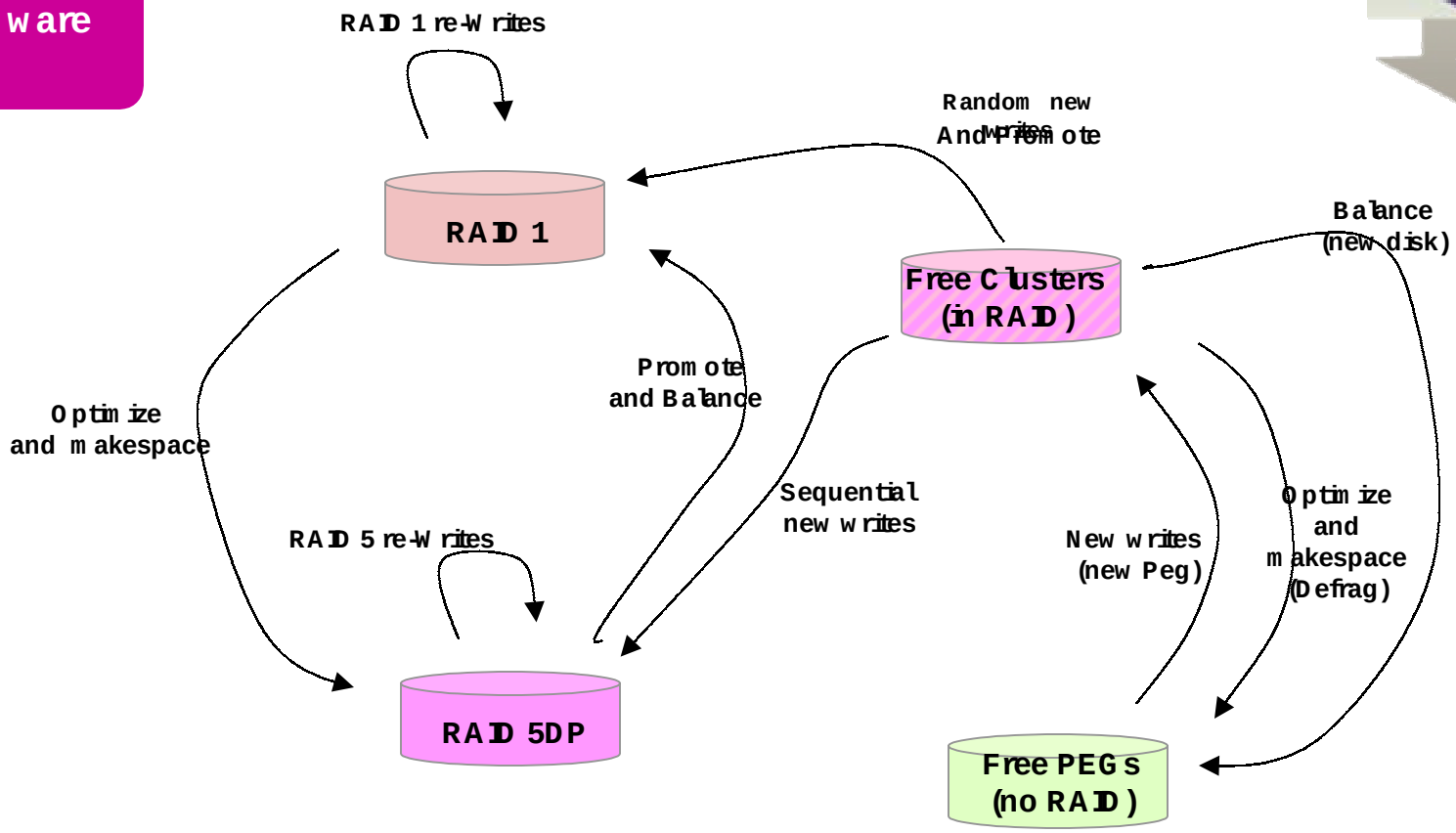
Sector – 512 bytes
RB/Cache Page – 128 sectors, 64K
Cluster – 4 RBs, 256K
PEX - 16 Clusters, 4MB
PEG – stripe of PEX's

level400 training

firmware

RAID state diagram

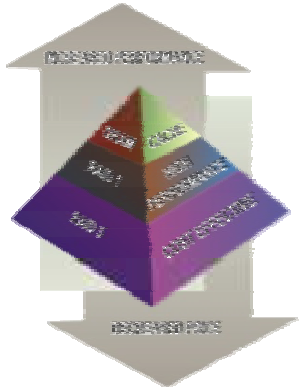




level400 training

firmware

data placement



- 'new writes' - assign free clusters to a LUN
  - RAID 1 if <64K write
  - RAID 5DP (stripe extending write) if:
    - LUN capacity will eventually need R5DP
    - >= 64K; Cache will gather smaller IOs into clusters
- re-writes (a previously assigned cluster)
  - always to same cluster (re-write in place)
- write time stamp - the controller 'remembers' when a write to a cluster occurs

RAID 5DP Stripe

16	52	5	10	18	P	Q

Stripe extending write, write new cluster to next unused cluster, P/Q are written once at end of stripe

Sector – 512 bytes
RB/Cache Page – 128 sectors, 64K
Cluster – 4 RBs, 256K
PEX - 16 Clusters, 4MB
PEG – stripe of PEX's

level400 training

firmware

makespace

Sector – 512 bytes
RB/Cache Page – 128 sectors, 64K
Cluster –4 RBs, 256K
PEX -16 Clusters, 4MB
PEG – stripe of PEX's

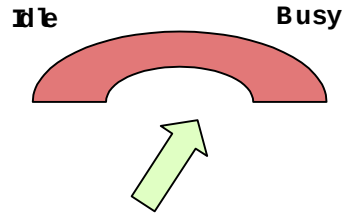
- makespace is called whenever there are no free clusters available for a 'new write' to complete
- makespace – moves data from RAID 1+0 to RAID 5DP, thus making more free clusters available
- foreground – must do prior to completing current hostIO
- selected first 'emptiest' PEG - not oldest cluster - optimized for selection speed
- moves enough R1 clusters to create an empty PEG

- *makespace is very disruptive to hostIO performance, because it must move a lot of data - it is to be avoided*

level400 training

firmware

array performance utilization calculation

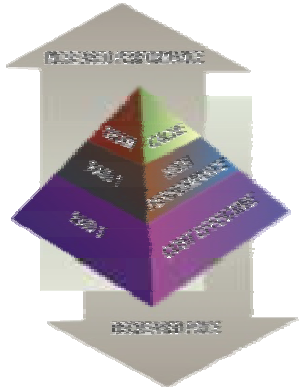


- the current array performance utilization is used as triggers for *optimize, promotion, rebuild, balance and scrub* - only under specific utilization will these policies run
- the controller collects backend workload statistics for the last minute of operation; reads, writes and data transferred
- utilization is computed (actual/max theoretical)
  - throughput (MB/s) utilization is the ratio of actual data transferred vs. 90MB/s (per RG)
  - transaction (IOPs) utilization is the ratio of actual IOs vs. computed capability of disk subsystem based on transfer size

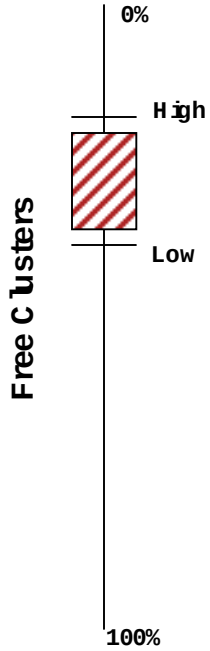
level400 training

firmware

optimize - make free space



- objective - create free PEGs, without effecting host IO workbad (to avoid makespace) - move RAID 1 to RAID 5DP
- checks status every two minutes
  - above high watermark:
    - Start background RAID 1 to RAID 5DP by oldest (write time) clusters first until high array performance utilization or below low watermark.
  - else if below low watermark
    - demote R1 clusters that are older than R5DP clusters (make room for later promotions)
  - then plugs holes (defrag) - makes PEGs both full and empty by consolidating partially filled PEGs



Sector – 512 bytes
RB/Cache Page – 128 sectors, 64K
Cluster – 4 RBs, 256K
PEX - 16 Clusters, 4MB
PEG – stripe of PEX's

level400 training

firmware

promote - active data to RAID 1

- move newer R5DP clusters to R1
  - only if space available
  - except data written to R5DP as full cluster
- stop at 10% R1 limit
- run only if array is in mid utilization

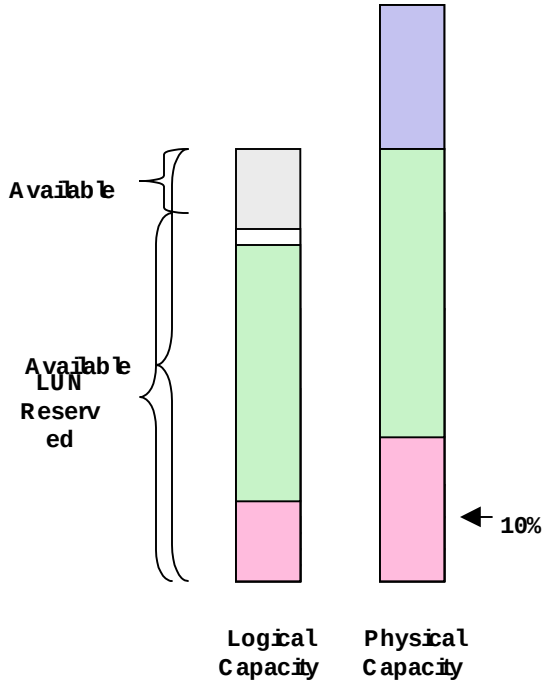
New for AutoRAID

- Deleting a LUN will cause promote to move reaming data to RAID 1
- Promote data only during low utilization times

Sector – 512 bytes
RB/Cache Page – 128 sectors, 64K
Cluster – 4 RBs, 256K
PEX - 16 Clusters, 4MB
PEG – stripe of PEX's

level400 training

firmware



1 - Physical disk capacity

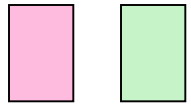
2 - less redundancy and free space gives usable

3 - create LUNs

4 -6 - writes

7 - optimize and promote

8 - more new writes



level400 training

firmware

performance statistics

- IO activity by disks (read and write)
- IO activity by array and LUN (read and write)
- new writes (R1 or R5DP)
- re-writes
- R1 and R5DP capacity (by LUN)
- R1 to R5DP movement
- R5DP to R1 data movement
- write working setsize

level400 training

firmware

balance - redistribute data to new disks

- triggered after new disk is inserted
- relocates all PEGs to new larger PEG
- RAID 1+0 first, then RAID 5DP
- if space available, promotion to RAID 1+0 on first PEGs available
- will suspend on moderate host workbad (50% utilization) or rebuild in progress

Balance objective:

- distribute data across new disks

- promote write active data to new RAID 1 space

Sector – 512 bytes
RB/Cache Page – 128 sectors, 64K
Cluster – 4 RBs, 256K
PEX - 16 Clusters, 4MB
PEG – stripe of PEX's

level400 training

firmware

rebuild - remove failed disk

- removes disk(s) from a RG by shrinking PEG size
- moves all data to new PEGs
- RAID 1+0 space first
- will only start if enough space to complete
- high priority
  - start next PEG move immediately after current completes
- low priority
  - start next PEG move after current completes if host workbad is < 25%, else
  - start next PEG move if > 6s since completion of last PEG

Sector – 512 bytes
RB/Cache Page – 128 sectors, 64K
Cluster – 4 RBs, 256K
PEX - 16 Clusters, 4MB
PEG – stripe of PEX's

level400 training

firmware

scrub - detect and repair latent media errors

- physical disk reads – independent of allocation
- 15 processes, independent of disk count
- read every sector, one at a time
- if read fails – log event
  - PDM will automatically recover and spare sector if needed
- suspend on low host activity (>20% utilization)
- if enabled – will run every 5 days, completion dependent on host activity
- 15 18GB disks – 2 hours

level400 training

firmware

cache

NVRAM per Controller	Write Cache	Read Cache	Total
256MB	100/50	24/12	124/62
512MB	200/100	52/26	252/126
1GB	600/300	150/75	750/375
2GB	1200/600	300/150	1500/750

7400/7100

- Separate read and write cache
  - Write in NVRAM, read isn't
  - Reads can be from write cache
- 64K cache page size
- Ifresiliency timer set
  - Testevery 4s
  - Cache pages with time olderthan 4s is queued forwrite - time setwith each write to cache page
  - Data is keptin write cache afterwrite, butmark as available
- Sequentialread ahead on 2nd miss in cache page (to end of cache page)
- Write ahead – queue many cache pages forwrite when above cache 'high water'mark

level400 training

firmware

AutoRAID improvements from Model12H

- optimize trigger was set to run only at very low performance utilization - thus it almost never ran and required makespace (very performance disruptive).
- no promote policy - promotion (R5 to R1) occurred only on write and one-at-a-time - thus write working set lagged R1.
- R5 did not have better sequential performance
- 20MB/s vs. 800MB/s memory bus; 1.6K vs. 30K cache IOPs
- rebuild priority works