



FLASH

REDEFINING THE POSSIBLE

Sam Marraccini
Flash Technology Evangelist
EMC Flash Products Division
Sam.Marraccini@emc.com
@SamMarraccini
www.INSIDEFLASH.com

Bart Sjerps
Advisory Technology Consultant - EMEA
Oracle, Business Intelligence & Data warehousing Solutions
bart.sjerps@emc.com
Blog: <http://bartsjerps.wordpress.com>
+31-6-27058830

Agenda

- EMC Flash Strategy
- Introduction – Oracle/EMC partnership
- Customer Challenges
- Performance best practices
- Flash Overview
- Hybrid Arrays
- Server Flash
- All Flash Array
- Questions

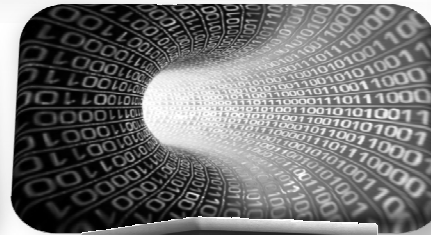
Mobile



Cloud



Big Data

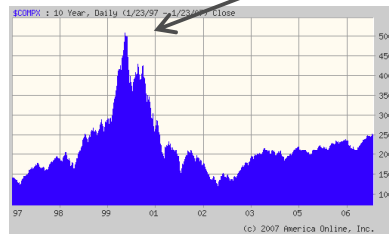


Social



TRUST

Memory Lane - Year 2000



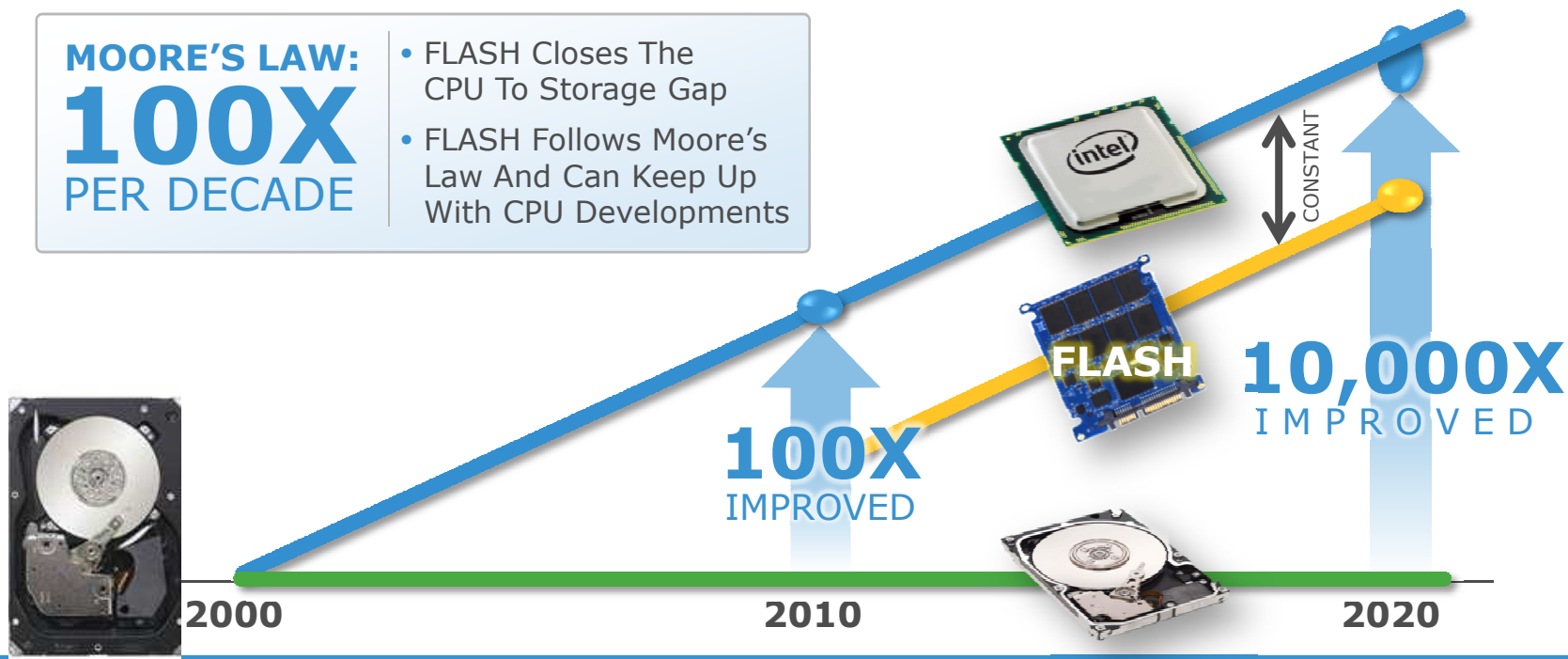
Seagate Introduces the 15K RPM Drive



IT Reality Check – CPU's vs HDD's

MOORE'S LAW:
100X
PER DECADE

- FLASH Closes The CPU To Storage Gap
- FLASH Follows Moore's Law And Can Keep Up With CPU Developments





Explosive
Data Growth

Bring Your
Own Device

Desperate Need
For Simplicity

Storage Is At The Center Of An IT Transformation

Why Not Now?

The Solid State Data Center **(not a matter of if, simply when)**

**BILLIONS
OF USERS**



**MILLIONS
OF APPS**



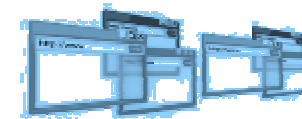
Mobile Cloud Big Data Social
Mobile Devices

**HUNDREDS OF MILLIONS
OF USERS**



LAN/Internet Client/Server
PC

**TENS OF THOUSANDS
OF APPS**



**MILLIONS
OF USERS**



Mainframe, Mini Computer
Terminals

**THOUSANDS
OF APPS**



Source: IDC, 2012

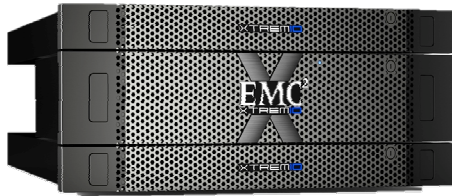


EMC Xtrem Product Portfolio

The Solid State Datacenter has arrived!

XtremIO

RETHINKING CORE DESIGN CRITERIA



Software-Defined
Off The Shelf Hardware

Inherently Balanced
Linear Scale-Out Architecture

Flash-Optimized
"Always On" Data Services

Scale iO

Elastic Converged Storage

Elastic Converged Storage

Elastic, Scalable & Resilient Shared Storage

XtremSW Cache

Performance, Protection & Intelligence
Caching Software extending

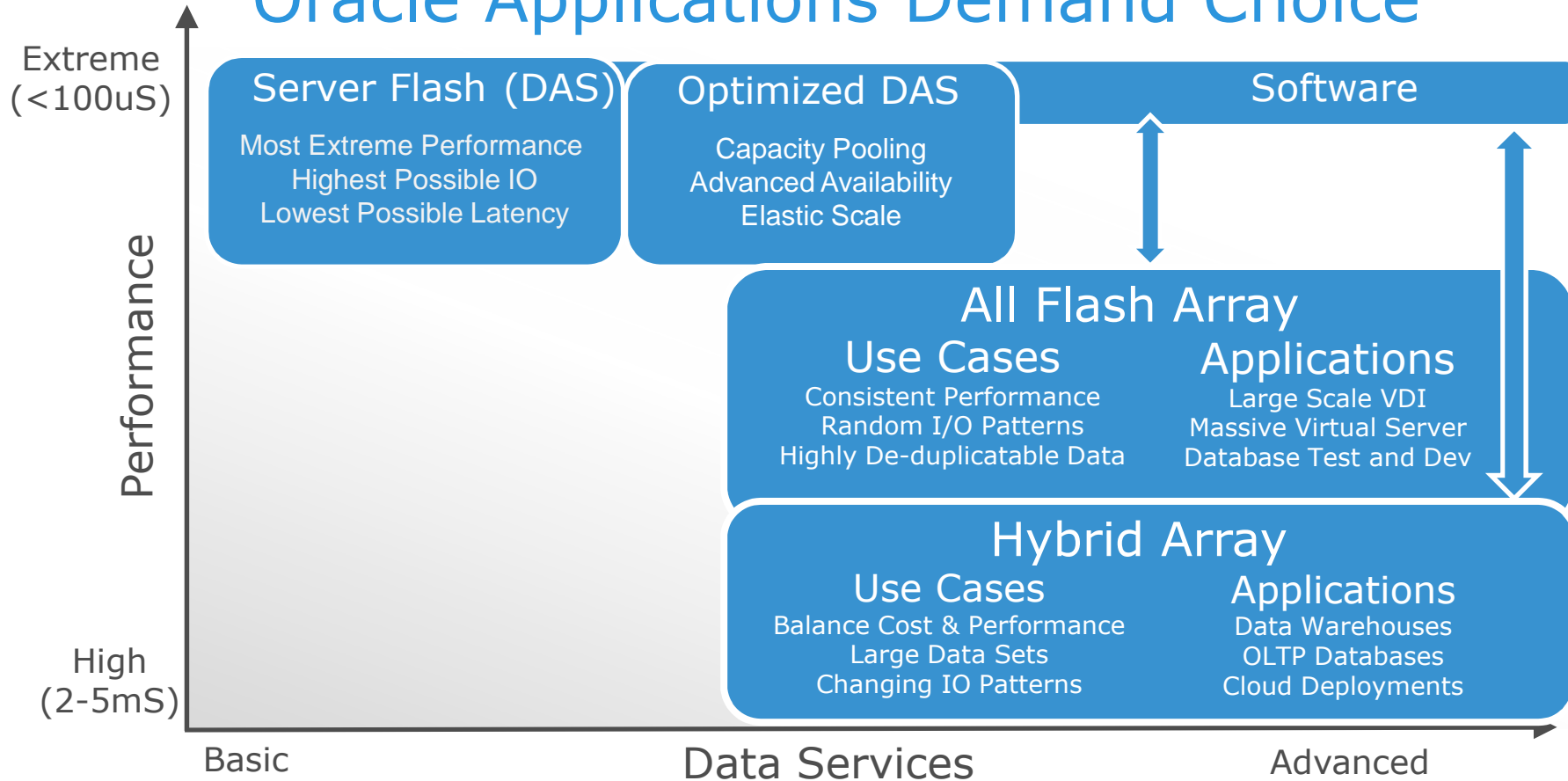


XtremSF

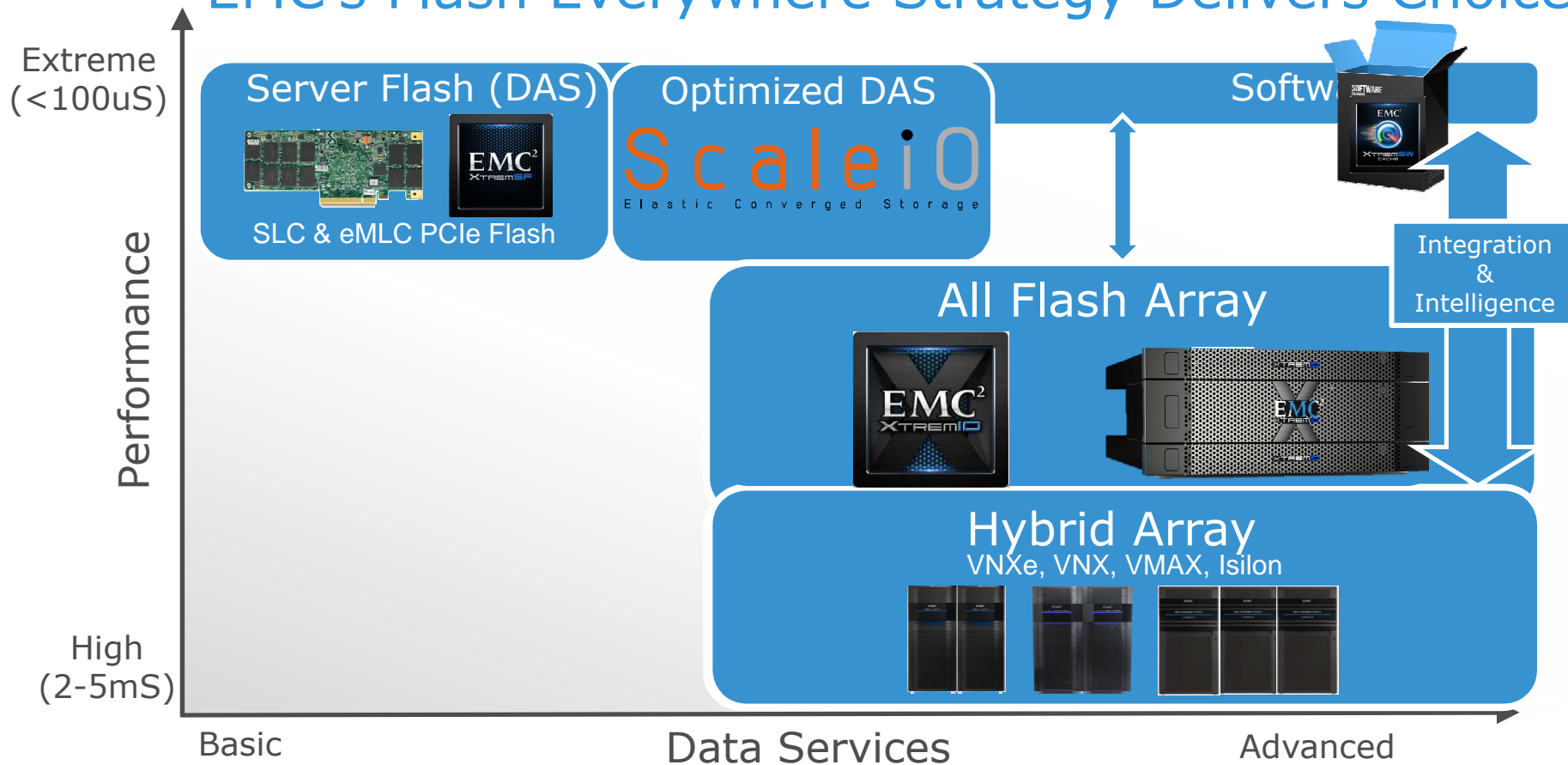
Server Side PCIe Flash
Highest Density in the Industry



Oracle Applications Demand Choice



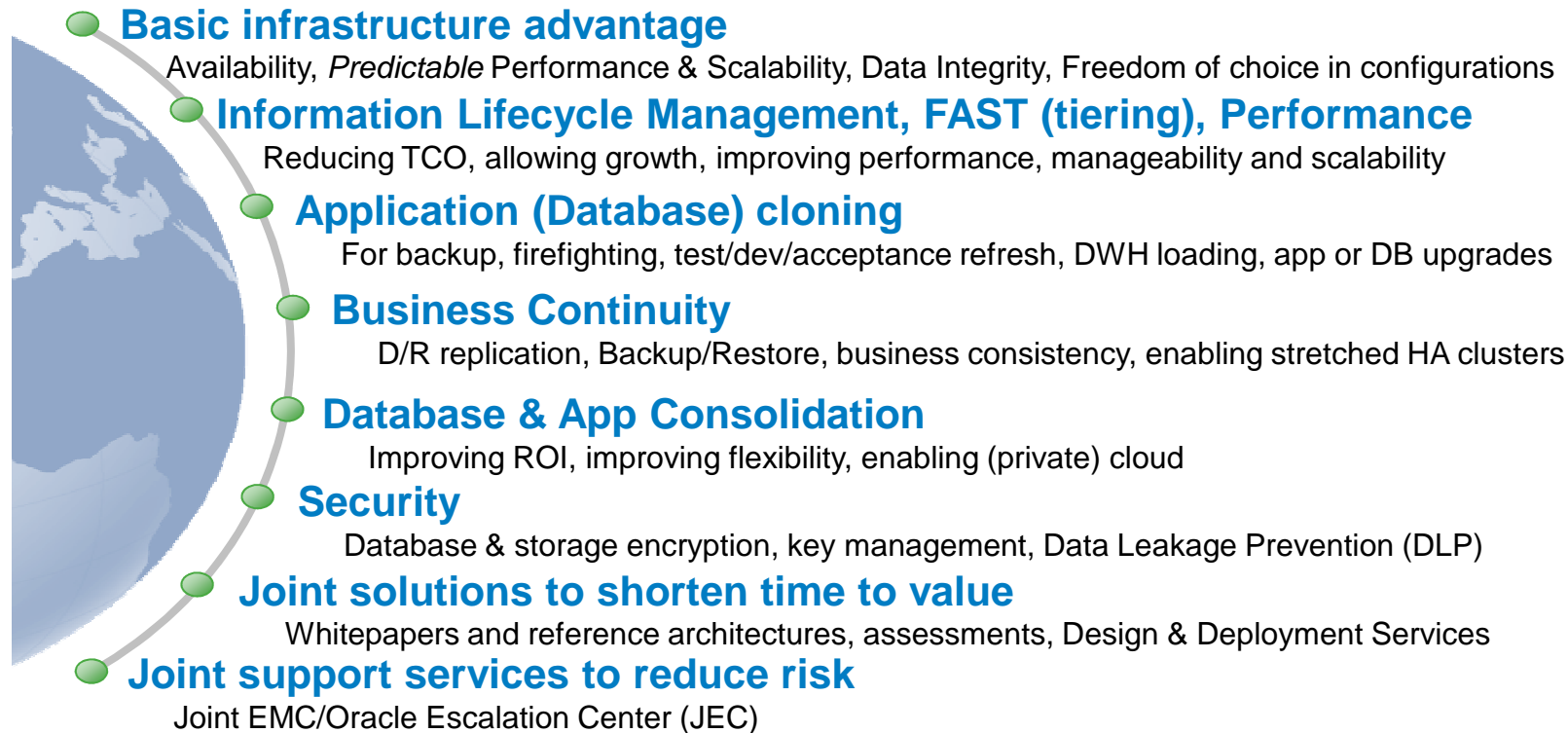
EMC's Flash Everywhere Strategy Delivers Choice



Why EMC for Oracle

ORACLE®

1995–Present



Customer challenges: Performance



- Still an issue after 40+ years of Moore's law
- Database sizes still grow
- As do workloads
- Applications don't always behave
- "Big Data" workloads

Findings from the field (1)

- DBA and storage teams don't always work well together
- Performance tuning focus on SQL and DB optimization
 - I/O underrated
 - Knowledge gap between DB and storage specialists
- Performance measured at different levels
 - But using deceptively similar metrics (i.e. response time)
- Best practices often not honored
 - Data layout, striping, block size, etc
- Limited performance tooling and capacity management in place

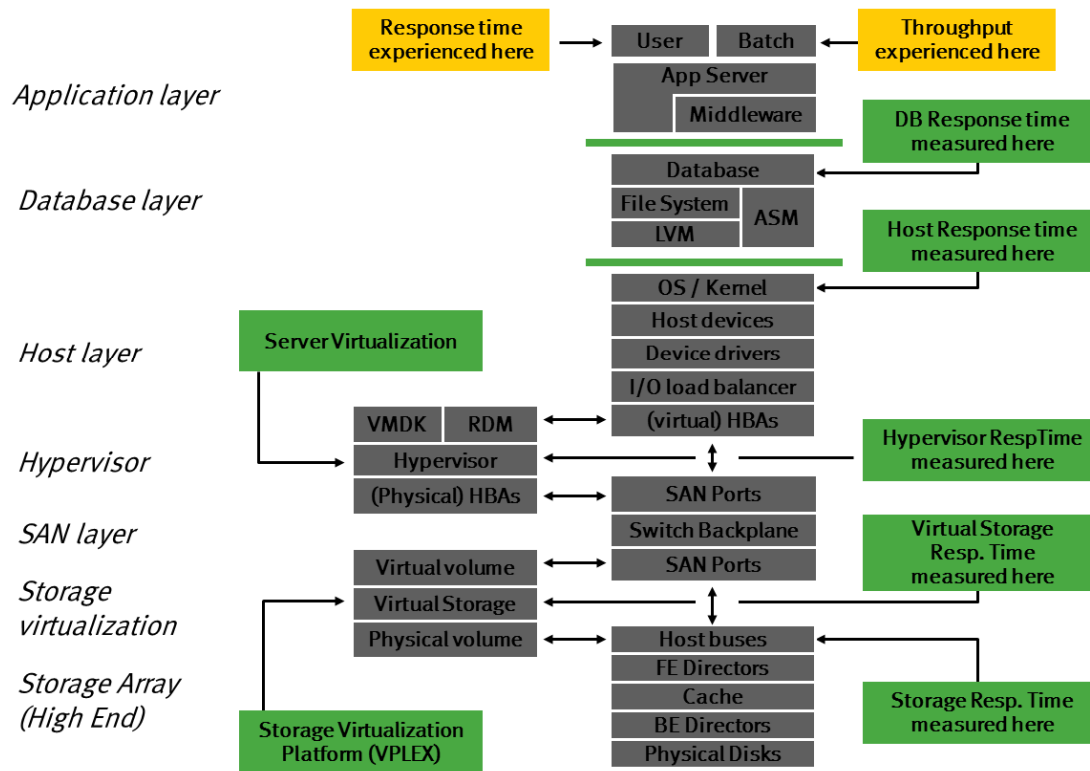


Findings from the field (2)

- Business expectations don't match IT
 - Undersized systems
 - Unexpected high peak loads
- Bottlenecks are not known
 - Adding CPU to avoid I/O problem
- Plain wrong architectural decisions
 - Limited up-front research, politics
 - [Conservative thinking](#)
- Storage as “black box”
 - “just give me my LUNs”
 - Ignoring storage characteristics such as striping, RAID, disk speed
 - Not using advanced storage features (i.e. snaps/clones, perf. features)



Understanding the whole stack



Users experience different performance than DBAs

DBAs measure different metric than storage admins (but named similar!)

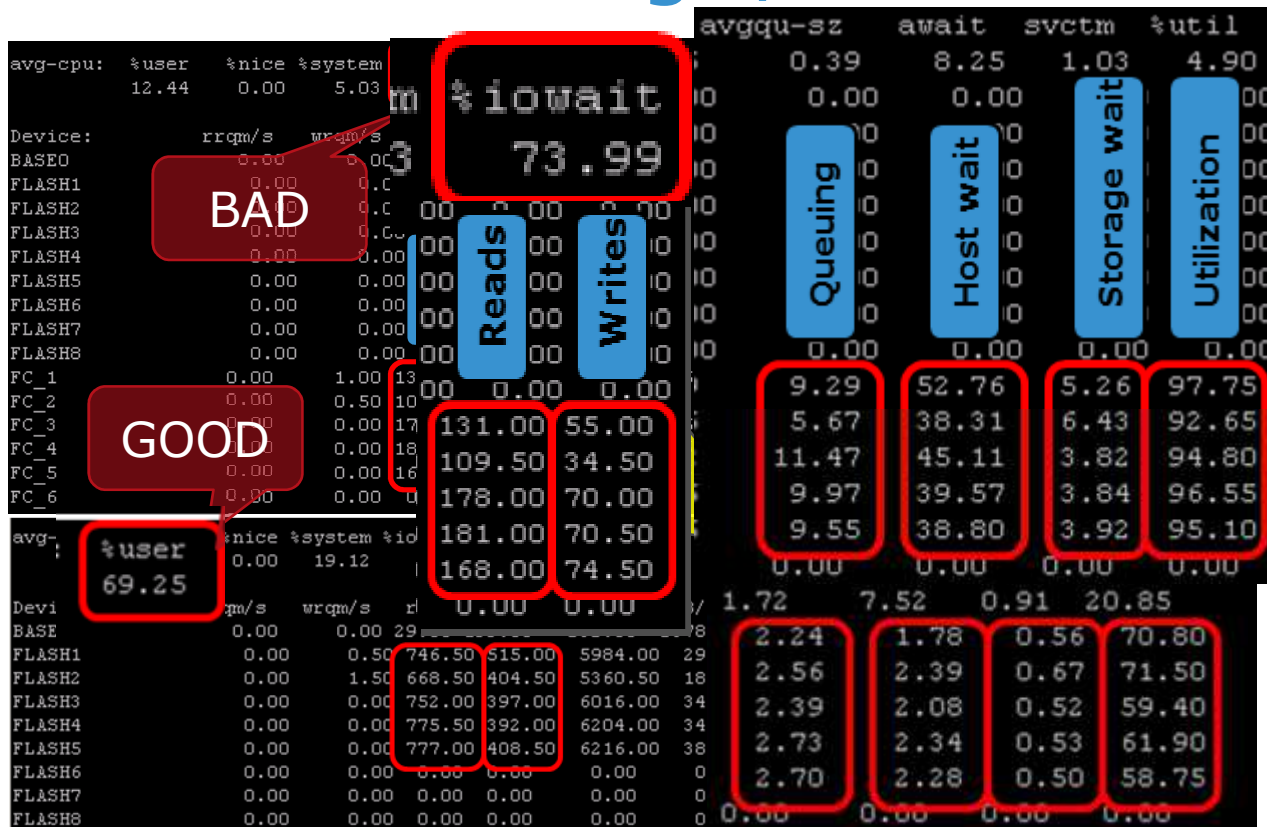
- If batch runs 2 hours, is that a perf issue?
- If CPU peaks 100%, is that a perf issue?
- If I/O wait is 95%, is that a problem?



Simplified overview of layers in the database stack: know what you're talking about



Understanding I/O wait



Linux:

`# iostat -xk 2 /dev/sdX /dev/sdY ...`

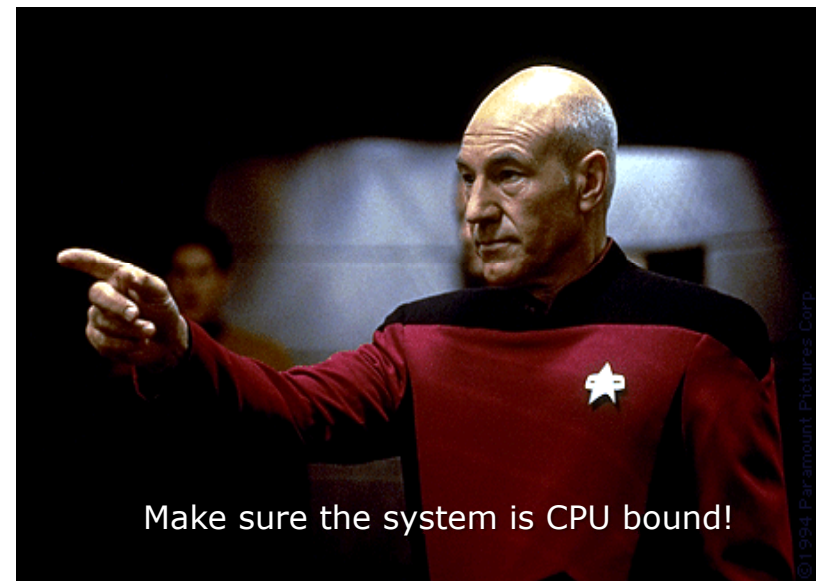
- Queuing happens (mostly) on the host
- Having multiple queues is common
- Utilization metric is unreliable

Goal:

Remove all I/O bottlenecks
CPU cycles are too expensive
to spend waiting. Or idling.

Databases shouldn't have high I/O wait

- Adding CPU does not speed up I/O bottlenecks
 - Memory does somewhat
- IOPS are relatively (!) cheap
- CPU cycles are expensive
- Databases have “hot” and “cold” regions
 - No need to make *all* storage fast
 - Modest amount of Flash will do – if applied correctly
 - Adding 5-10% Flash can boost performance by over 80%
 - YMMV 😊



Locality of reference



- Oracle was developed in a time where CPU and memory was expensive (thus limited)
 - Disks perform well (both read and write) if you avoid disk head movements (seeks)
 - How many IOs per sec do you get from cheap SATA disk – given *sequential* 8K reads?
 - Therefore database stores related data as close together as possible
- Locality of reference

Oracle Database I/O behavior

- Reads are not always sequential but short sequences and related I/O may happen, i.e. block offsets 1001 → 1002 → 997 → 1004 → 1005 → 1009 (consider B-tree index, range scans)
- Storage caching algorithms can optimize this. Consider all of these blocks share a physical disk track – if we do a seek to get to 1001 let's then read the whole track in cache. Now the first I/O (1001) has 7ms resp. time, the rest has << 1ms ☺
 - Since 1995, EMC has invested heavily in R&D (i.e. analyze I/O traces etc.) to improve these algorithms
 - Note that tablespace and file system fragmentation, striping and other indirection mechanisms (Volume managers, write-anywhere file system schemes) can ruin your day ☹
- If you have sequential write data it could make sense to assign dedicated disks
 - REDO logs, DWH staging areas

I/O skewing



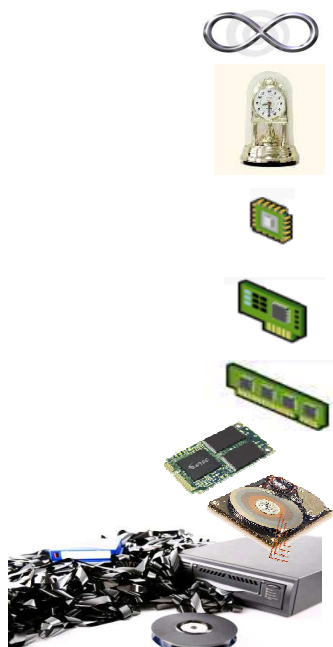
- Database objects (indexes, tables) tend to grow by appending blocks at the end
- Due to the nature of business processing, the most recently added data (rows) are likely to be retrieved more often
- The oldest data is less likely to be very active
- So we get (slowly moving) hot spots (and respectively, cold spots) in the data
- This is called "skewness" i.e. 80/20 skew means 80% of I/O happens on 20% of the data blocks
- In that case you can reduce seek time on 80% of all I/O requests to be below 1ms – by putting it on FLASH storage (but the devil is in the details)

Flash versus Spinning Disk

Spinning disk	Flash Disk (SLC / eMLC)
One operation at a time	Parallel operations – any workload
Mechanical movements required for seeks	No mechanical parts
Cannot handle high utilization well	High utilization is fine
Reads perform like writes – no need for zero out before write	Writes require clearing out flash regions first – sustained writes may cause degraded perf
Sweet spot: sequential R/W	Sweet spot: random read
I/O directly relates to physical offset on disk	I/O offset obfuscated due to wear leveling
Typical resp. time ~ 7 ms	Typical resp. time ~ 0.5 ms
Random IOPS ~ 150	Random IOPS ~ 3000 (depends!)
Bandwidth ~ 70 MB/S (sequential read/write)	Bandwidth ~ 70 MB/s (sequential read)
Wears out by age, not usage	Wears out by (overwrite) usage
No wear leveling required	Needs wear leveling
Requires caching algorithms for (random) performance	Requires caching algorithms for (write) performance + endurance

Access times of storage media

Typical relative speeds of components (2013) 1ns = 1s

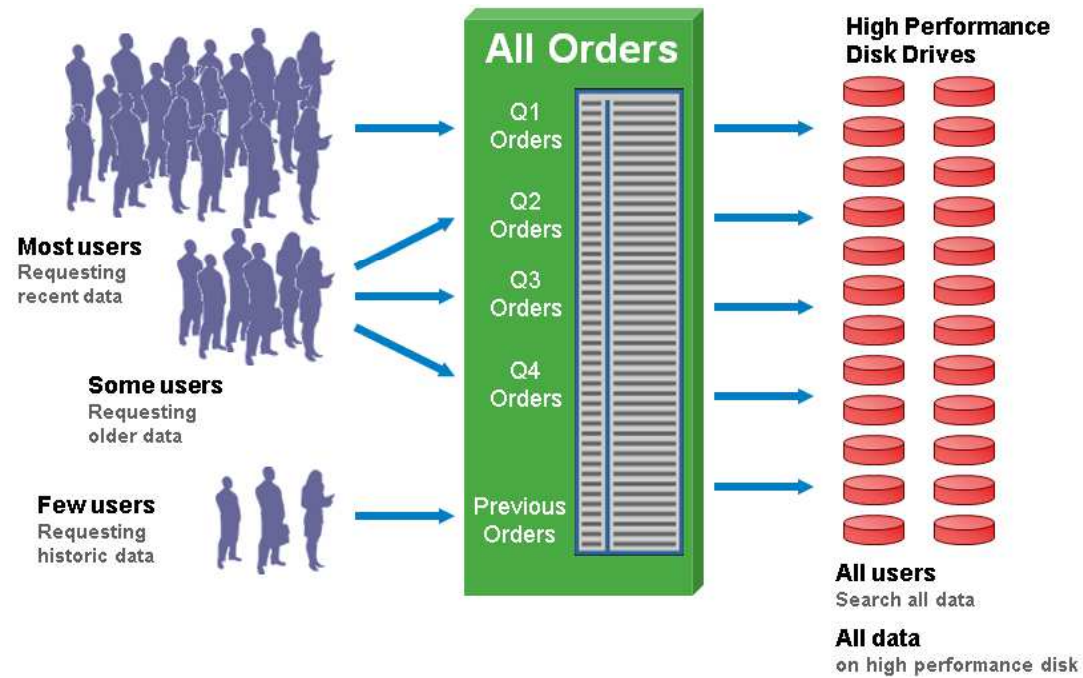


Access type	Typical Cycle Time (nanoseconds)	Cycle time (s)	Scaled Cycle Time (scale = 10 ⁹)	Typical Capacity
Avoided IO	Zero	Zero	Zero	-
CPU clock (2.5 GHz)	0.4	4 x 10 ⁻¹⁰	0.4 seconds	-
L1 cache	2	2 x 10 ⁻⁹	2 seconds	64KB
L2 cache	4	4 x 10 ⁻⁹	4 seconds	256KB
L3 cache	25	25 * 10 ⁻⁹	25 seconds	4 MB
DRAM	100	100 x 10 ⁻⁹	1 minute 40 sec	256 GB
PCIe Flash	50,000	50 x 10 ⁻⁶	14 hours	1 TB
Flash Disk	500,000	0.5 x 10 ⁻³	5 days	10TB
Rotating Disk	7,000,000	7 x 10 ⁻³	3 months	100TB
Tape	10,000,000,000	1 x 10 ⁺¹	3 centuries	Petabytes

↑ Capacity ↓
ISO

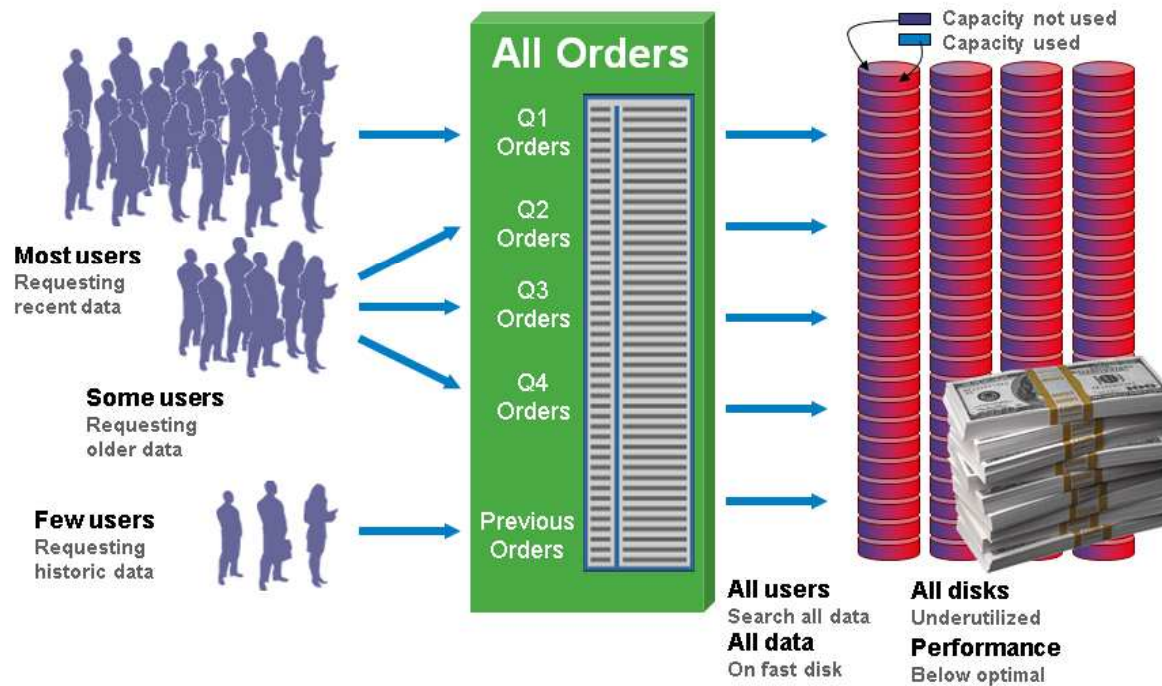
Database Storage Tiering (ILM)

Traditional deployment



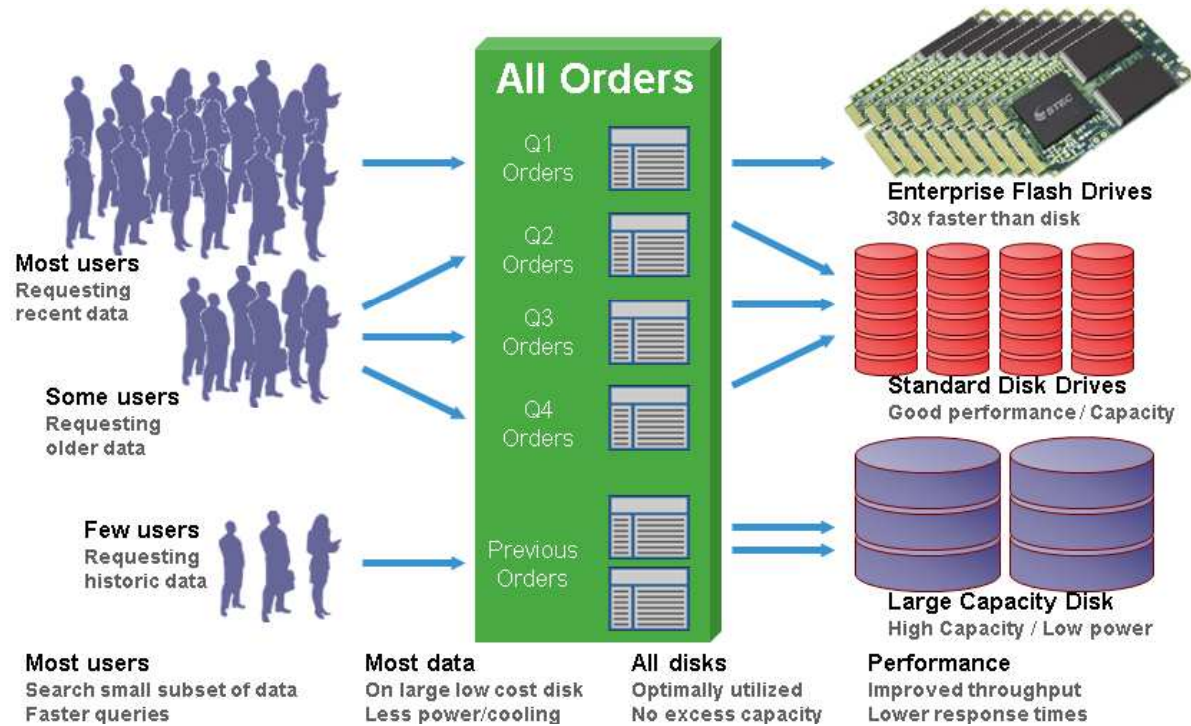
Database Storage Tiering (ILM)

Growing database sizes



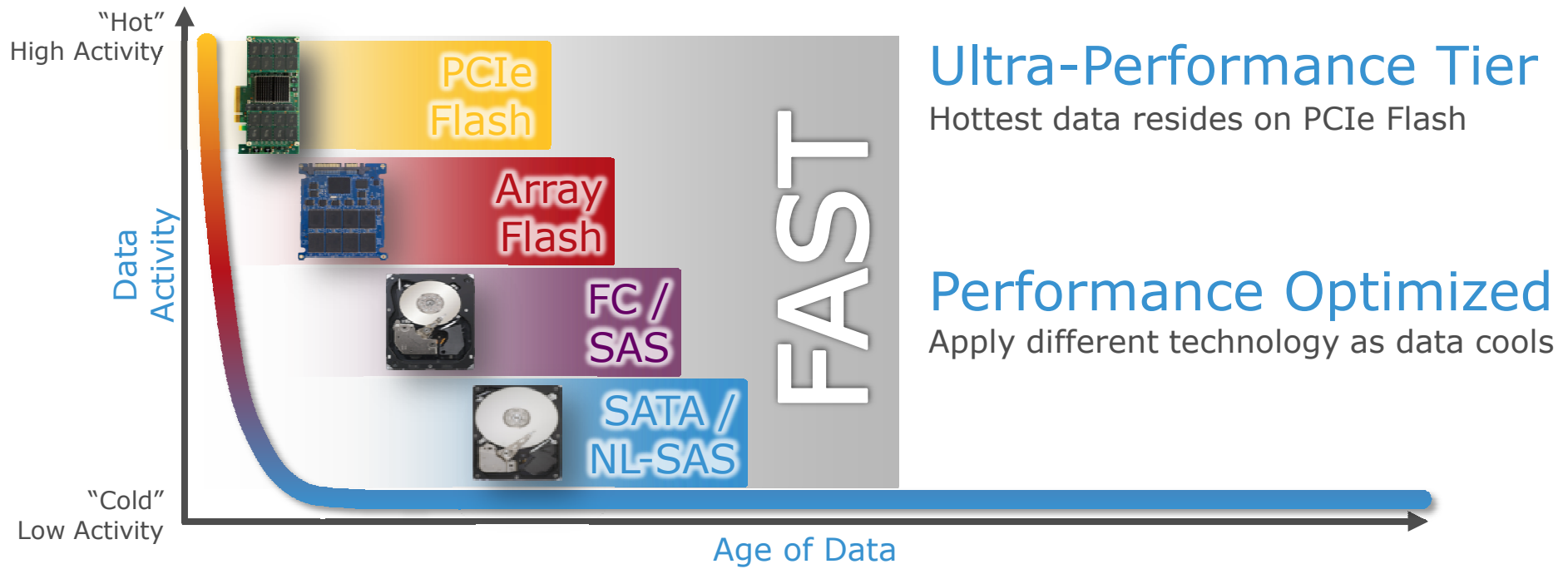
Database Storage Tiering (ILM)

Implementing ILM

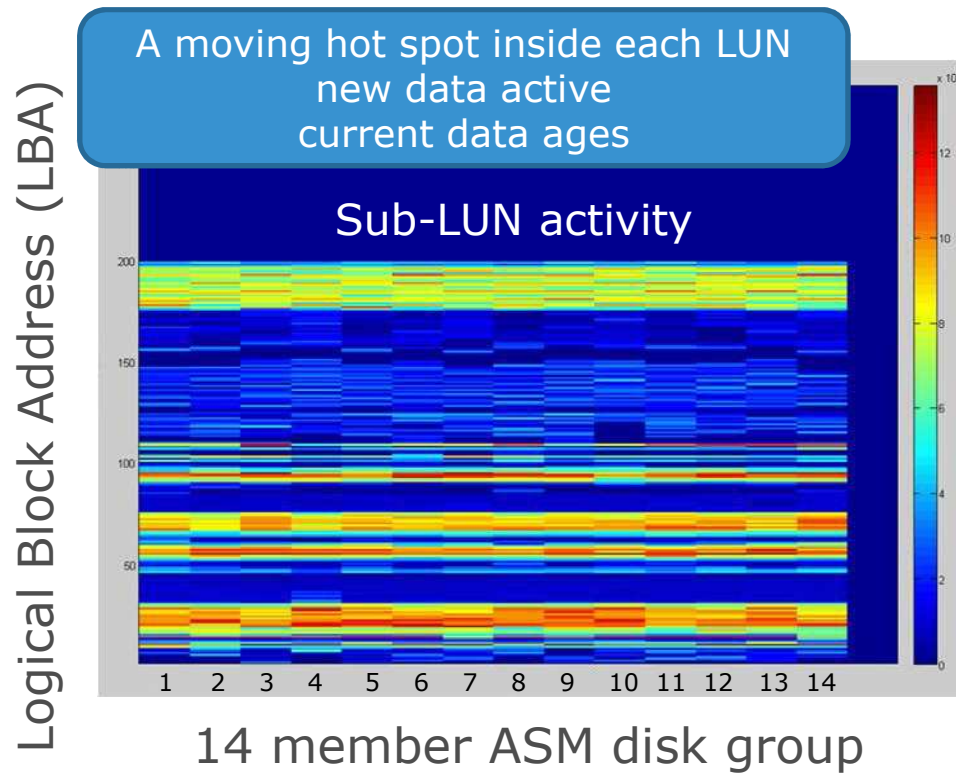


Automated Intelligence

EMC Fully Automated Storage Tiering (FAST VP)

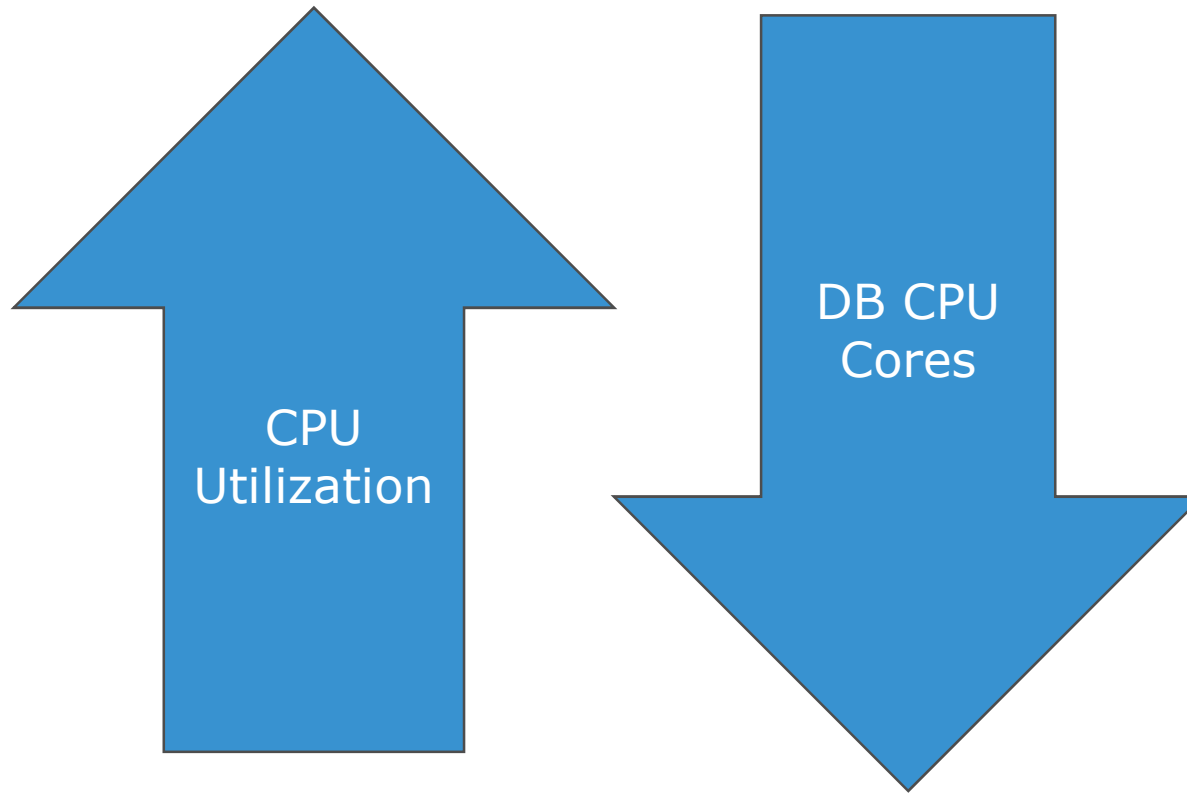


FAST VP And Oracle Databases



- Oracle DB tends to stripe across many LUNs
 - Using LVM striping (like ASM) or using many filesystems and db files
- Users tend to access the most current data
 - Small portions of data are very active and keep changing over time
- Optimization for cost/performance
 - Full ILM strategy (complex/tedious)
 - Let FAST VP do it!
- FAST VP and Sub-Lun Tiering
 - Works very well with Oracle databases and ASM

The Big Payback



Best Practices (1)

EMC recommends various settings for good performance.

Examples:

- Linux Hugepages
 - Reduces CPU overhead in managing Linux memory management
- Linux I/O scheduler
 - Elevator or deadline? Or CFQ?
- Queue depths
 - Tradeoff between response time and throughput
- EMC Powerpath for load balancing
 - Works better than native or 3rd party “MPIO”-style balancers
 - Linux MPIO is known to sometimes chop large I/O into 4K chunks (bad)

Best Practices (2)

- Disk alignment
 - Use 64K or 1MiB (both are fine)
 - Linux “fdisk” creates 31,5K “misaligned” partitions – resulting in overhead
 - More info: <http://bartsjerps.wordpress.com/2013/03/28/linux-alignment-reloaded/>
- REDO logs
 - 100% sequential write
 - No duplexing required unless 3rd party vendors require this (has no benefit for protection)
 - Don’t make larger REDO log groups than needed
 - “External redundancy” - EMC is very good at data protection, don’t spend precious host CPU and I/O cycles on that
 - Where possible, dedicate physical disk groups for REDO. RAID-5 FC/SAS is fine. Sharing with other DBs is fine.
 - Where possible, dedicated I/O channels might reduce response times (avoid REDO IO having to wait for background DB writer I/O for example)

Best Practices (3)

- Striping
 - Oracle 11.2: defaults to coarse striping for REDO. Change back to FINE striping (128K)
 - Avoid striping for everything else (both ASM and FAST-VP avoid hotspots anyway)
 - Really avoid double striping (can kill all prefetch / performance algorithms)
- ASM
 - Separate ASM disk groups
 - Increase default ASM AU size to $\geq 8\text{MB}$ (recommended 16MB)
 - REDO logs, FRA/ARCH, TEMP and regular data files
 - Sometimes it makes sense to go beyond that and split some index/data
- TEMP
 - Create TEMP on dedicated FLASH/EFD if DB uses TEMP for sorting/joining etc
 - TEMP generates random read/write which is boosted by using Flash

Best Practices (4)

- Remote Replication
 - Asynchronous SAN replication typically has ZERO performance impact but still guarantees consistency
 - And reasonable RPO for many applications (~ 5 to 10 minutes)
 - Use SYNC only where really needed (such as financial processing)
 - No matter if you use Data Guard or SAN replication (i.e. SRDF, Recoverpoint)
- Database init parameters
 - Don't modify things for performance POCs that you wouldn't modify in production
 - Such as block checksum "disabled" settings and other exotic stuff
 - We're in search of realistic, predictable, not just "breaking the record" performance numbers
 - DB block size: 8KB (DWH benefits from $\geq 16K$ sometimes). Never go lower than 8K !

Best Practices (5)

- Queue depths
 - Large queue depth: more throughput
 - Small queue depth: better response time
 - No silver bullet / single recommendation
- Consistent, predictable “good” performance is better than unpredictable, unreliable “Guinness World Records” performance
 - Can athletes consistently achieve world records? Or once in a lifetime?
 - So test performance also under “special conditions”
 - Such as disk failures, broken cables/channels, during RAID rebuilds, with SYNC replication enabled (i.e. Data Guard or EMC SRDF), when performing DB cloning using snaps/clones, when users are submitting crazy table scans, ...
 - During backups / restores (same server or same cluster / shared infra)

Best Practices (6)

- Oracle RAC?
 - Can sometimes cause more problems than improvements due to RAC interconnect traffic
 - A workload that requires 30 CPU cores is typically better off with a 32-core single-node server than a 2-node 16-core/node cluster
 - Use when you need extreme availability (mostly not performance as large single-node servers do better)
 - In that case, consider Oracle RAC stretched clusters (with [EMC VPLEX](#))
 - Generic HA (cluster) tools can offer quick failover times as an alternative
 - And don't forget license cost
- Beware of CPU Overhead
 - Specific hypervisors: 4% (as measured by EMC IT)
 - Oracle RAC: no hard numbers (but most would agree it's at least 10%)
 - Host replication (i.e. ASM redundancy, log shipping): ~ 1-2% CPU + mirrored writes
 - Don't run anything else on DB server except DB processing! (No apps, middleware, mgt agents, ...)

Best Practices (7)

- IP based protocols
 - (Direct) NFS as good as Fiber Channel these days
 - Provided one applies all best practices (jumbo frames, non-blocking switches, 10GigE, ...)
 - Excellent alternative to ASM, dNFS = 100% NFSv3 compliant (no vendor-specific magic)
- Exotic filesystems?
 - Avoid ZFS for primary datafiles (heavy fragmentation and other issues, requires lots of tuning)
 - Avoid OCFS/OCFS2 (performance, I/O chopping™ into 4K, not mainstream)
 - Other filesystems: YMMV ;-)
 - Be prepared for lots of “Evil” tuning of bottlenecks
 - Filesystems use RAM that otherwise could be allocated to SGA
 - And prefetch less efficient than DB itself
 - Beware of heavy memory paging / thrashing

RAID levels & disk types for Oracle datafiles

- Data / Index
 - Read and Write
 - Large & small I/O
 - Both Random & sequential
 - RAID-5 is OK, RAID-1 is (a bit) better
 - Avoid RAID-6 (and RAID-6 - like)
 - Split tablespaces if you need to squeeze out that extra 5%
 - Isolate from REDO, ARCH, FRA, etc on physical disk level
 - A bit of FLASH a day keeps the performance doctor away
 - Auto-tiering (FAST-VP)!
- REDO logs
 - 100% sequential write
 - RAID-1 or RAID-5 (both are OK)
 - No need for 15K rpm (but use this if rest of system also uses 15K)
 - FC/SAS is OK (no need for EFD/Flash)
 - Preferably on dedicated physical disks (if redo I/O is high)
 - Sharing with other databases is fine
 - Tune for fast write response times of small block I/O
 - Exclude from tiering

RAID levels & disk types for Oracle datafiles

- Binaries
 - Any (reliable) storage is OK
- TEMP
 - Separate if high DB TEMP usage
 - Very random I/O pattern (if used)
 - Used for joins / sorts / aggregates
 - And Index builds (+ reorg?)
 - On Flash/EFD where needed
 - Regular tier is OK if no high TEMP usage (shared with DATA)
- FRA/ARCH
 - Confusion: used for both Archive logs and backup files, and Flashback logs...
 - All three are good candidates for RAID-6 SATA (cost-effective) as performance is not very important
 - Sometimes contains control files as well (tricky with replication) – avoid!

REDEFINE THE POSSIBLE DATABASES





EMC CAN HELP YOU **LEAD YOUR** TRANSFORMATION

Sam Marraccini
Flash Technology Evangelist
EMC Flash Products Division
Sam.Marraccini@emc.com
@SamMarraccini
www.INSIDEFLASH.COM

Bart Sjerps
Advisory Technology Consultant - EMEA
Oracle, Business Intelligence & Data warehousing Solut
bart.sjerps@emc.com
Blog: <http://bartsjerps.wordpress.com>
+31-6-27058830

EMC²®