High Performance solutions for Oracle

Experiences & best practices

Bart Sjerps | Principal Systems Engineer | Oracle Specialist - EMEA
bart.sjerps@dell.com | +31-6-27058830
http://bartsjerps.wordpress.com

DELL EMC

THE NEW STORAGE AUDIENCE: DBA

Oracle DBAs

Being Asked To Do More…

Performance

Availability

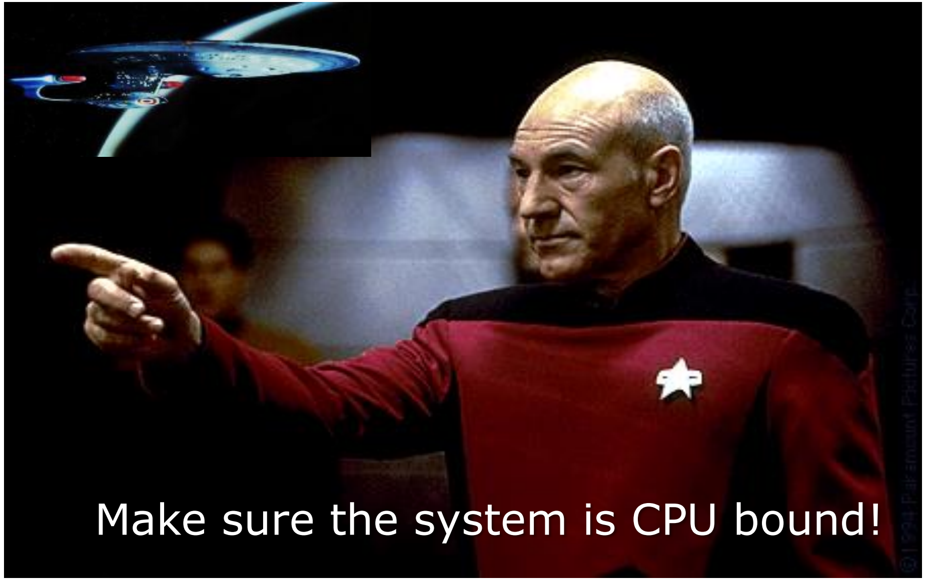
Management

Storage Admin

Being Given More Tools…

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
  - Because of licenses
- 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 ☺



STORAGE IS NO LONGER THE BOTTLENECK

EMC²

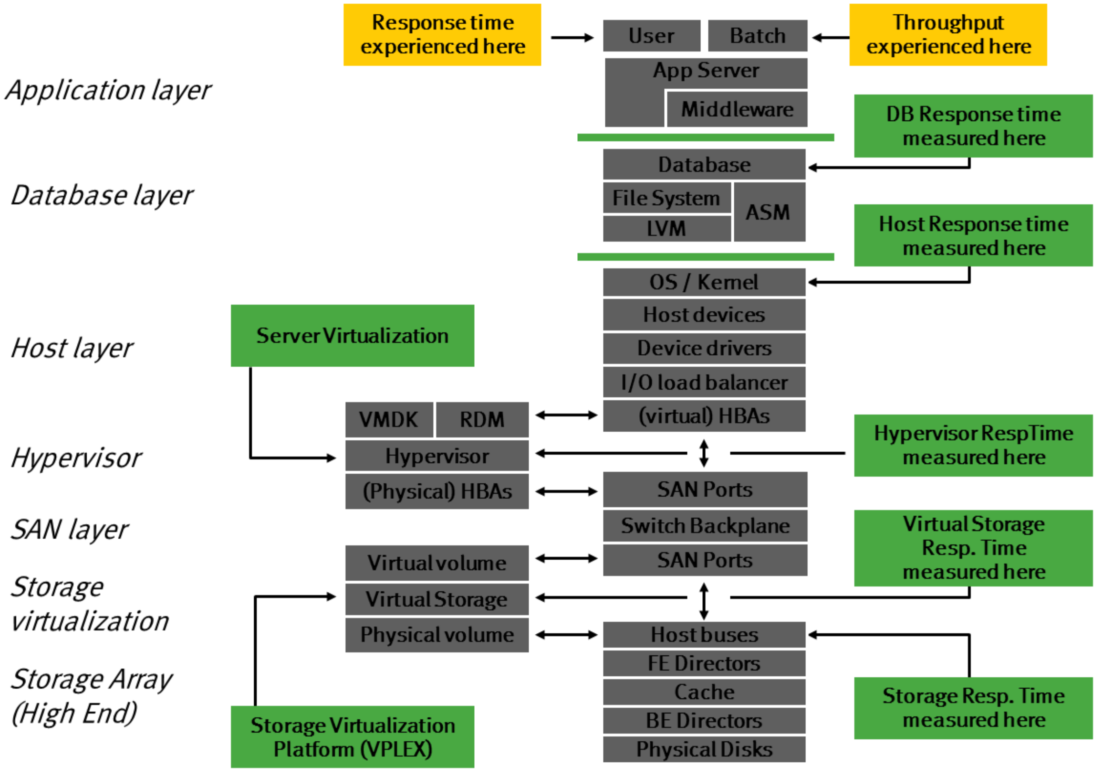
Findings from the field (1)

- DBA and storage teams don't always work well together
- Performance tuning focus on SQL and DB optimization
  - I/O and storage are underrated
  - Knowledge gap between DB and storage specialists
- Performance measured at different levels
  - But using deceivingly similar metrics (i.e. response time)
- Best practices often not honored
  - Data layout, striping, block size, alignment 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"
  - As per the myth told by storage vendor marketing/sales (including EMC…) "the new hardware is so fast, doesn't need tuning"
  - Ignoring storage characteristics such as striping, RAID, disk speed
  - Not using advanced storage features (i.e. snaps/clones, performance features)
  - SATA is cheap, let's put everything on large RAID-6 SATA disks!

UNDERSTANDING THE WHOLE STACK



Users experience different performance than DBAs

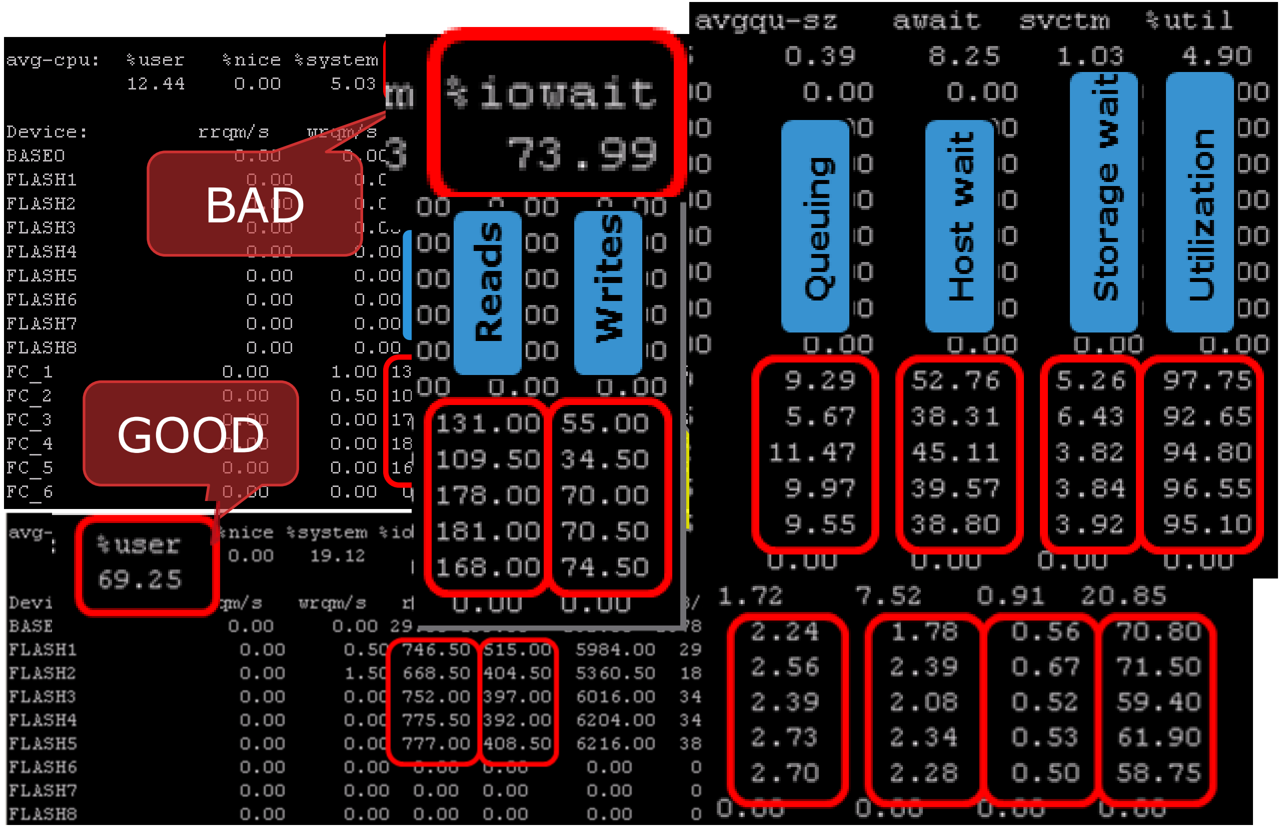
DBAs measure different metric than storage admins (but named similar!)

- 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

EMC²

UNDERSTANDING I/O WAIT



Linux:

 # iostat –xk 2 /dev/sdX /dev/sdY …

$$\textit{Host Wait} = \textit{Service time} + \textit{Queuing time}$$

- **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.

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)

The Performance Gap Challenge

CPU Improves 100 Times Every Decade; HDD Remains Flat

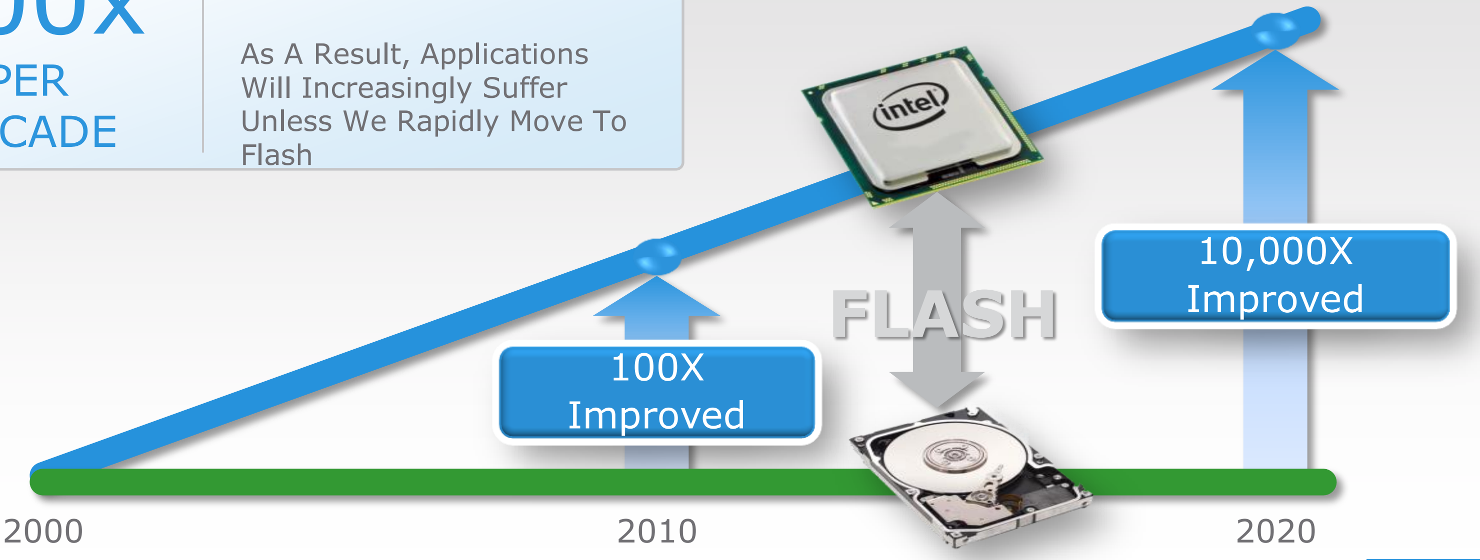
MOORE'S LAW:

100x

PER DECADE

CPU Continues To Improve While Disk Drive Performance Remains Flat

As A Result, Applications Will Increasingly Suffer Unless We Rapidly Move To Flash



FLASH VERSUS SPINNING DISK

Single spinning disk	Single 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 performance - Garbage collection required
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 (@ low % busy)	Typical resp. time ~ 0.5 ms (@ high % busy)
Random IOPS ~ 150	Random IOPS ~ 3000 (depends!) (* outdated)
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 good (random) performance	Requires caching algorithms for (good write) performance + endurance

EMC

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	4×10^{-10}	0.4 seconds	-
L1 cache	2	2×10^{-9}	2 seconds	64KB
L2 cache	4	4×10^{-9}	4 seconds	256KB
L3 cache	25	$25 * 10^{-9}$	25 seconds	4 MB
DRAM	100	100×10^{-9}	1 minute 40 sec	256 GB
Flash Memory	50,000	50×10^{-6}	14 hours	1 TB
Flash Disk	500,000	0.5×10^{-3}	5 days	10TB
Rotating Disk	7,000,000	7×10^{-3}	3 months	100TB
Tape	10,000,000,000	$1 \times 10^{+1}$	3 centuries	Petabytes

⇧ COST
Capacity ⇩

ORACLE ON EMC BEST PRACTICES

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?
  - Virtual: NOOP!
- Queue depths
  - Tradeoff between response time and throughput
  - No good "formula" available. Trial & error.
- EMC Powerpath for load balancing
  - Consistent over all platforms, fire & forget
  - Works better than native or 3rd party "MPIO"-style balancers
  - Linux MPIO is known to sometimes chop large I/O into 4K chunks (bad)

ORACLE ON EMC BEST PRACTICES

- 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
  - ASM: 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)

EMC²

ORACLE ON EMC BEST PRACTICES

- 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
  - External redundancy!
  - Separate ASM disk groups
  - Increase default ASM AU size to ≥ 8MB (recommended 16MB)
  - Split 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 storage

ORACLE ON EMC BEST PRACTICES

- 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)
  - ZERO Dataloss is (partly) a myth: The Zero Dataloss Myth blogpost
  - No matter if you use Data Guard or SAN replication (i.e. EMC 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 ≥ 16K sometimes). Never go lower than 8K !
  - Many parameters that potentially influence IO (such as MBRC)

ORACLE ON EMC BEST PRACTICES

- 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?
  - Should we 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)
  - During firmware updates

ORACLE ON EMC BEST PRACTICES

- Oracle RAC?
  - Can sometimes cause more problems than improvements due to RAC interconnect traffic, locking, pinging etc
  - 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
  - These days a single Intel host can have 80+ processors. Why scale out? Scale up!
  - 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: VMware ESX overhead= 4% (as measured by EMC IT)
  - Oracle RAC: no hard numbers (but many 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, …)

ORACLE ON EMC BEST PRACTICES

- 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, see my blogposts on the matter)
  - Avoid OCFS/OCFS2 (performance, I/O chopping™ into 4K, not mainstream)
- Other filesystems: YMMV ;-)
  - Be prepared for lots of "Evil" tuning of bottlenecks
  - Filesystems often use RAM that otherwise could be allocated to SGA (use directIO etc)
  - FS prefetch is much less efficient than DB caching itself -> disable!
- 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 policies

RAID LEVELS & DISK TYPES FOR ORACLE DATAFILES

- Binaries
  - Any (reliable) storage is OK
- TEMP
  - Oracle's "paging space"
  - 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!

PERFORMANCE PROOF OF CONCEPTS

PROFILING NEW SYSTEMS BEFORE YOU GO LIVE

- Always test low-level performance
  - Using a mix of “dd”, “iorate” or Vdbench, etc
- Always test transactional workloads
  - Using Swingbench, HammerDB or similar TPC-C “like” tools
- Always test IOPS and throughput
  - Only one tool is good enough: SLOB
  - SLOB does IOPS only (random read and/or write)
  - “slob-fulltablescans.sql” adds sequential read (bandwidth) test (beware: single threaded for now): Slob Full table scans (blogpost)
- Only after basic tests, run your own custom queries
- Now you’re confident to go live ☺

EMC²

SUGGESTED WORKLOAD GENERATING TOOLS

- Swingbench
  - Has become the De-facto tool to simulate OLTP workloads
  - Swingbench SOE (Sales Order Entry) has become the "unofficial" TPC-C like benchmark
  - Typically CPU bound (if infra configured to have no I/O bottlenecks i.e. use Flash where needed etc)
  - Performance may vary depending on generated data size and DB configuration (i.e. SGA, block size etc) – the detailed DB stack configuration + Swingbench setup must be documented and repeated across different tests
  - Not a good tool to drive lots of I/O
  - Very good tool to compare CPU power of platforms
  - Note that OLTP is often CPU-bound (like many DWH queries for that matter)
- SLOB
  - The "Silly Little Oracle Benchmark" created by Kevin Closson
  - Not a real benchmark but a pure Oracle I/O generator
  - Basically generates lots of database block reads and/or writes (plus redo I/O) without driving high CPU
  - Use it to profile I/O limits without depending on CPU and memory
- UNIX tools
  - dd, cp, etc: good for getting initial "feel" if the system is driving enough bandwidth
  - Not a good benchmark
- IORate
  - EMC public domain tool to generate I/O (without database)
  - Can be used for initial profiling
  - If all works well, should match SLOB results (more or less)

PERFORMANCE POC SUGGESTIONS

- Use both Swingbench and SLOB
  - Swingbench to profile TPC-C like transactions per minute
  - SLOB to profile I/O workload
- Test multiple workloads (different servers) at the same time
- Using VMware CPU shares, see how service levels are met
  - i.e. a VM “prod” with 2000 shares should get more TPM than a VM “test/dev” with 500 shares if they share the same physical host
  - See if and how VMware starts moving workloads across physical servers to balance out the workloads real-time
- Test the replication to physical server procedure
  - Oracle might occasionally ask for that when providing support
- Optional: Using and auditing CPU affinity
  - To manage license cost in some occasions

DELL EMC