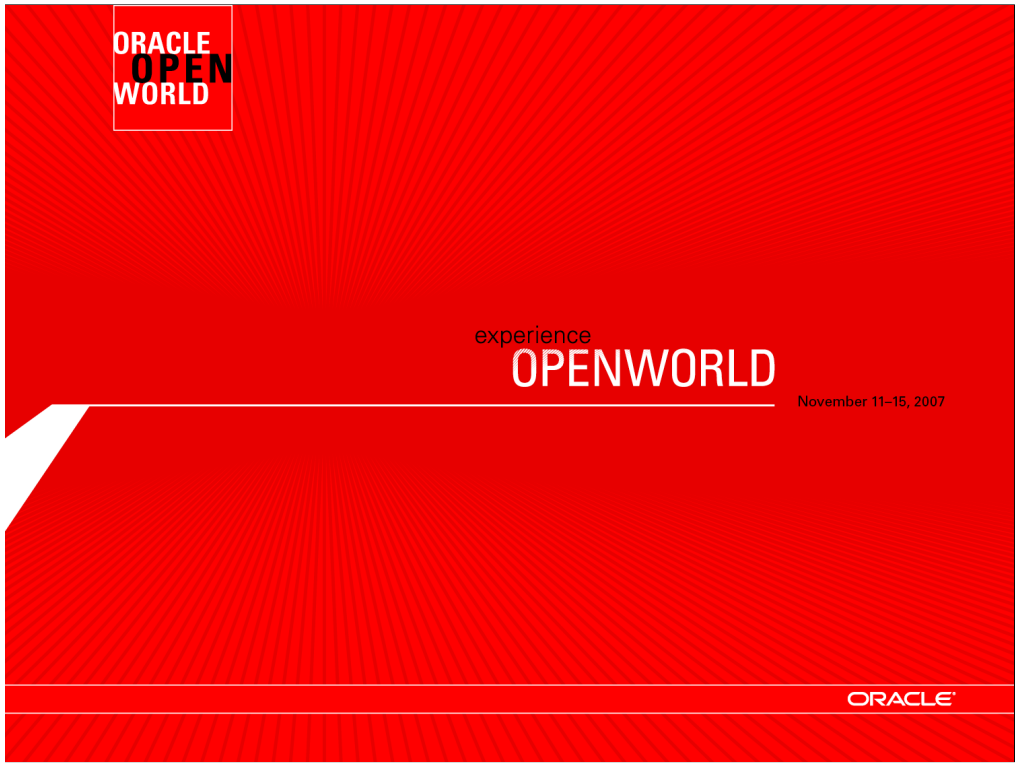

ORACLE


ORACLE
OPEN
WORLD
experience
OPENWORLD
November 11–15, 2007
ORACLE

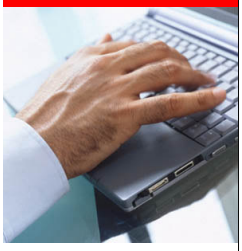
ORACLE

RAC PACK
Back-of-the-Envelope Database Storage Design

Nitin Vengurlekar
RAC/ASM Development

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions.
The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Program Agenda



- Review key Terms and Definitions
- I/O Design Methodology
- Overview of I/O components
- ASM Overview and Impact
- Piecing together the picture
- Validating the Configuration
- Summary

ORACLE

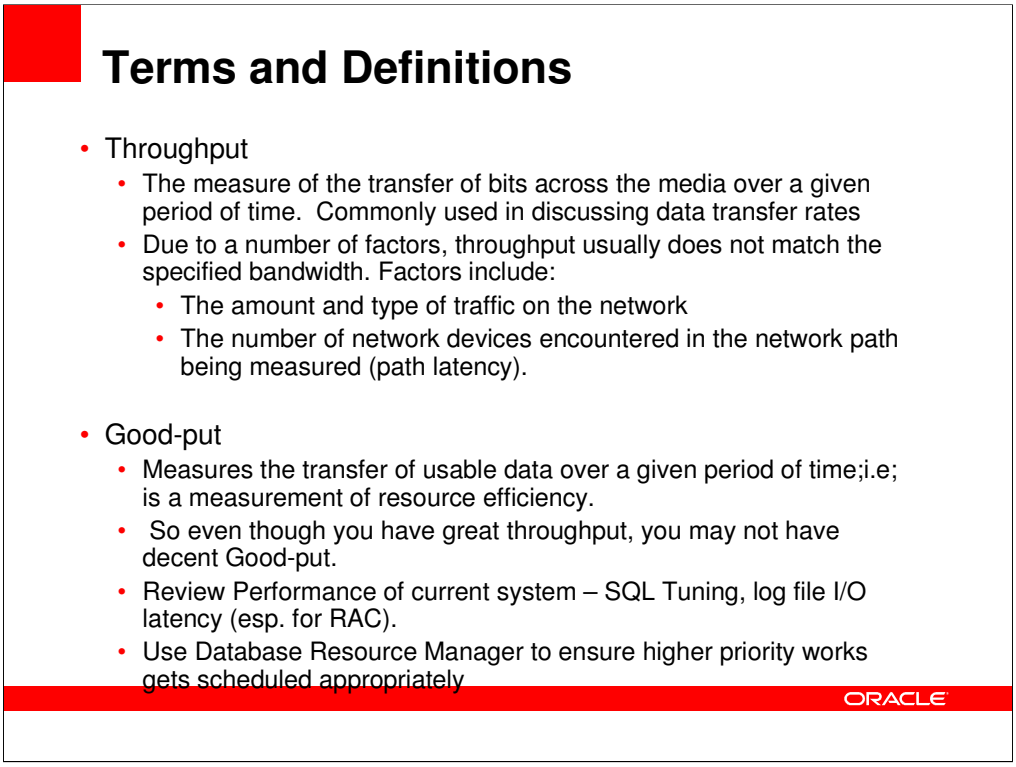
Session Objective

- Understand how to define your I/O infrastructure to meet your application requirements
  - Proactive not reactive
- Provide enough supporting information to communicate effectively with Storage team
  - Use common terms

ORACLE

Although this presentation though can be applied to iSCSI or SAS networks, its primarily for FC subsystems, since they're the most predominant.

The primary purpose of this session is to provide DBAs with enough supporting information to intelligently speak and communicate with their Storage Admin, so that application I/O requirements are properly conveyed and understood by the entire group.



Throughput

Throughput is the measure of the transfer of bits across the media over a given period of time. Due to a number of factors, throughput usually does not match the specified bandwidth.

Many factors influence throughput. Among these factors are the amount and type of traffic and the number of network devices encountered on the network being measured (path latency).

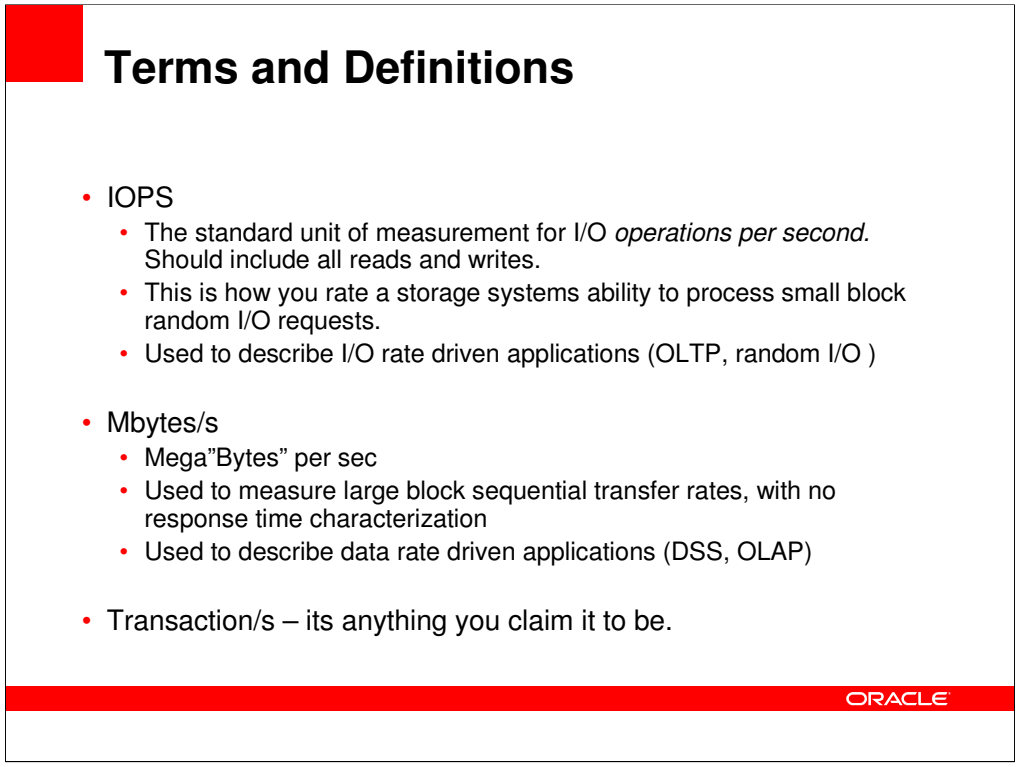
In any network (including FC networks), throughput cannot be faster than the slowest link of the path from source to destination. Even if all or most of the segments have high bandwidth, it will only take one segment in the path with low throughput to create a bottleneck to the throughput of the entire network.

Bandwidth is the amount of information it is physically possible to send through the media of choice

Goodput

Review, executions per cursor, rows per execution, etc.. Bad goodput will not scale well.

Measure the transfer of usable data. That measure is known as goodput. Goodput is the measure of usable data transferred over a given period of time. So even though you have great throughput, your goodput may not be that well enough. This is because your queries are not well tuned (though they may good good response times), and thus request more data than is necessary. Goodput is a measurement of resource efficiency.



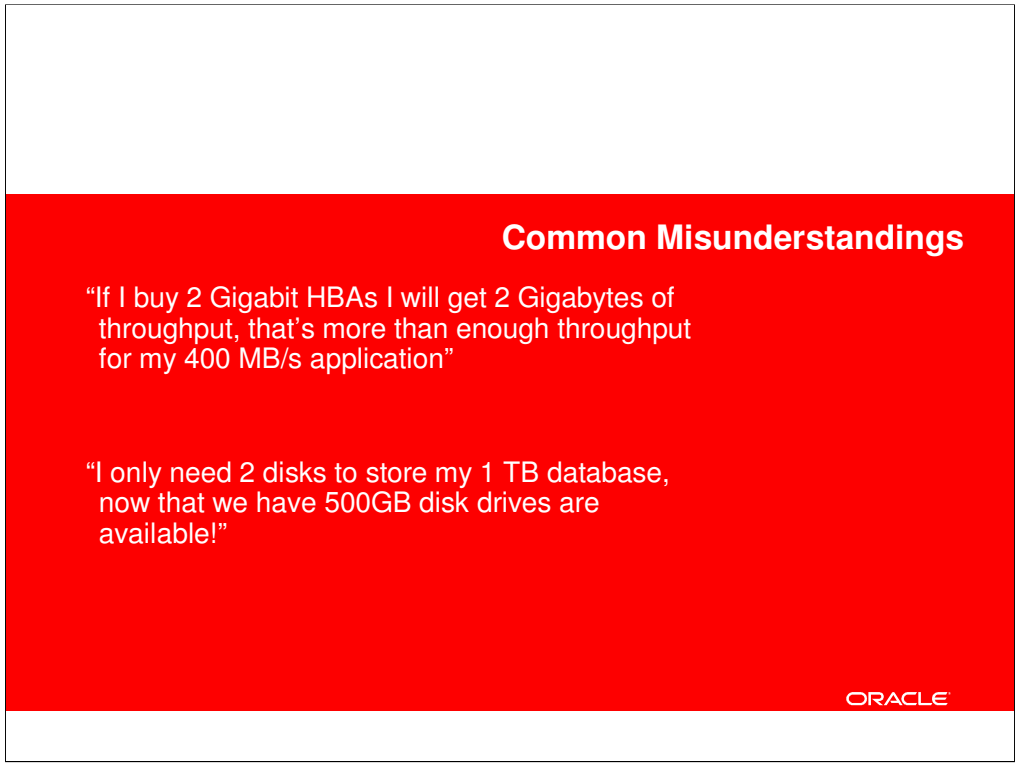
You can use the following from CERN to aggregate some of this data. Thanks to CERN folks on putting this together.

```
set lines 250
    set pages 9999
    spool sysmetric_outp.log

    alter session set nls_date_format='dd-mm-yyyy hh24:mi';

    select min(begin_time), max(end_time),
        sum(case metric_name when 'Physical Read Total Bytes Per Sec' then average end)
    Physical_Read_Total_Bps,
        sum(case metric_name when 'Physical Write Total Bytes Per Sec' then average end)
    Physical_Write_Total_Bps,
        sum(case metric_name when 'Redo Generated Per Sec' then average end)
    Redo_Bytes_per_sec,
        sum(case metric_name when 'Physical Read Total IO Requests Per Sec' then average end)
    Physical_Read_IOPS,
        sum(case metric_name when 'Physical Write Total IO Requests Per Sec' then average end)
    Physical_write_IOPS,
        sum(case metric_name when 'Redo Writes Per Sec' then average end) Physical_redo_IOPS,
        sum(case metric_name when 'Current OS Load' then average end) OS_LOad,
        sum(case metric_name when 'CPU Usage Per Sec' then average end)
    DB_CPU_Usage_per_sec,
        sum(case metric_name when 'Host CPU Utilization (%)' then average end) Host_CPU_util, --
    NOTE 100% = 1 loaded RAC node
        sum(case metric_name when 'Network Traffic Volume Per Sec' then average end)
    Network_bytes_per_sec,
        snap_id
    from dba_hist_sysmetric_summary
    group by snap_id
    order by snap_id;

    spool off
```



To convert 2Gbits into Mbytes :

2Gbits = (2* 1024 * 1024 * 1024) / (1024* 1024) / 8bits

= 2147483648 / 1048576

= 2048 / 8

= 256Mbytes

I/O Design and Planning

- Typical scenarios for I/O design and planning
  - Building a new system from scratch – new infrastructure
  - Growing the existing application – extend/augment current infrastructure
    - We'll focus on this scenario for the session

ORACLE

I/O Design and Planning

- Determine the application I/O characteristics
  - RPO, RTO, response time SLA, IO rates, etc.
- Understand each I/O component's bandwidth limits
- Choose the appropriate I/O components that will match application requirements
- Goal:
  - Design for throughput not for capacity
  - Design for scalability and availability

ORACLE

RPO – recovery point objective ; the point/state that data needs to be recovered to.

RTO – recovery time objective ; the time required to allowed recovery of data.

Determine Application I/O characteristics

- Use AWR reports to determine I/O metrics (Instance Activity Stats per sec).
  - IOPS = “physical reads total I/O requests” + “physical writes total I/O requests”
  - MBytes/s = “physical reads total bytes” + physical writes total bytes”
  - For RAC environments - aggregate IOPS or MBytes/s for all nodes
- Include Backup Requirements
  - Define amount of data to be backed up, frequency and time allotted for backup window
- These values and the application characteristics should be communicated to the System & Storage Administrators.

ORACLE

You can pull these stats using:

```
set lines 250
set pages 9999
spool sysmetric_outp.log

alter session set nls_date_format='dd-mm-yyyy hh24:mi';

select min(begin_time), max(end_time),
      sum(case metric_name when 'Physical Read Total Bytes Per Sec' then average end)
       Physical_Read_Total_Bps,
      sum(case metric_name when 'Physical Write Total Bytes Per Sec' then average end)
       Physical_Write_Total_Bps,
      sum(case metric_name when 'Physical Read Total IO Requests Per Sec' then average end)
       Physical_Read_IOPS,
      sum(case metric_name when 'Physical Write Total IO Requests Per Sec' then average end)
       Physical_write_IOPS,
      snap_id
from dba_hist_sysmetric_summary
group by snap_id
order by snap_id;

spool off
```

For new or non-existing applications, use business rules or data model transaction profiles flow to understand “what is a transaction”, and then extrapolate for transactions/s or per hour. Optionally you can use the numbers we have seen in our consulting gigs. Note that these are just guideline values

Use the following as basic guidelines for OLTP systems :

Low transaction system – 1000 IOPS or 200MBytes/s

Medium transaction system – 5000 IOPS or 600 Mbytes/s

High-end transaction system – 10,000 IOPS or 1Gbytes/s <- almost rarely achievable and usually TPC-C type workloads

Use the following as basic guidelines for DSS systems (units are in Gig = Gigabytes/Sec):

Customer example - AWR report

Instance Activity Stats Per second stats

physical read total IO requests	1,197.43
physical read total bytes	67,932,081.00
physical write total IO requests	1,050.86
physical write total bytes	65,114,880.37

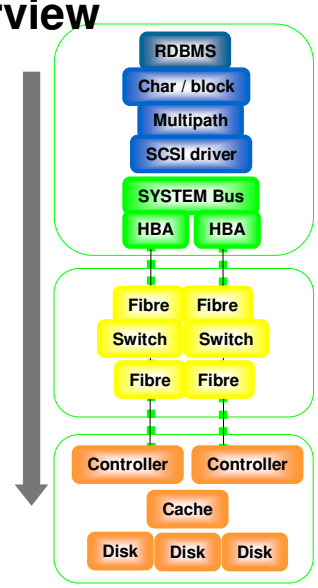
Total:
physical read total I/O requests + physical write total IO requests ~ 2247 IOPS
physical read total bytes + physical write total bytes ~ 133 Mbytes/s

ORACLE

Once the IOPS and MBytes/s are determined, the “heavy lifting” is done.

I/O stack components Overview

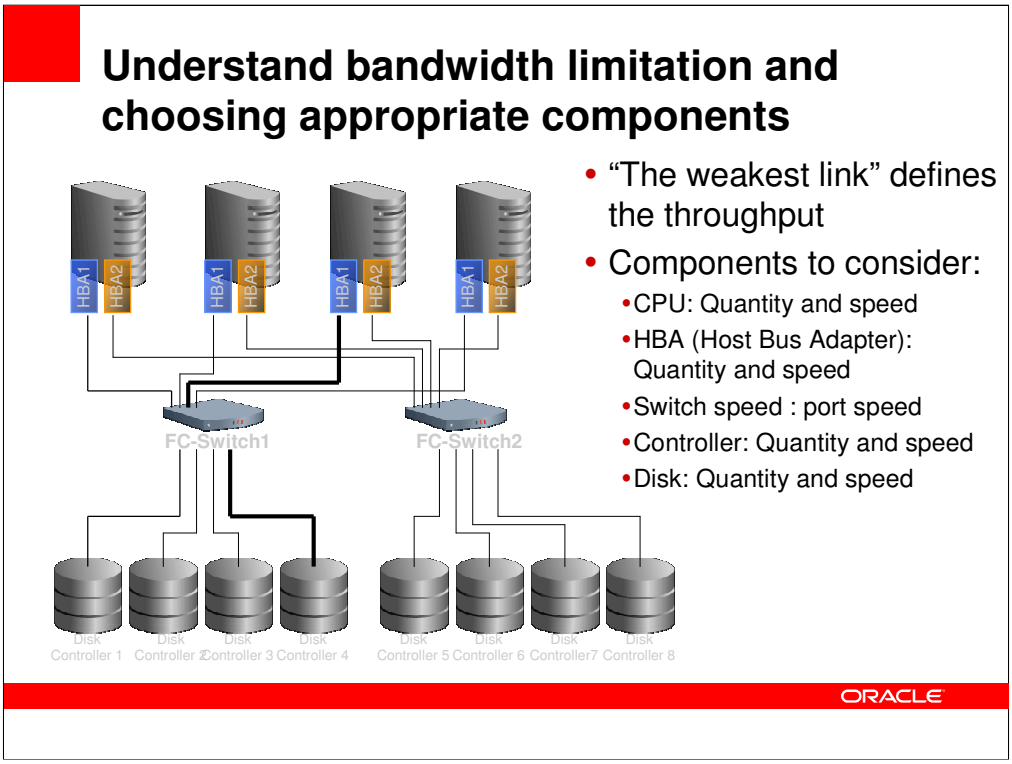
- Database
- Character or block device driver
- Multipathing driver
- SCSI driver
- HBA
- System Bus
- Switch/iSCSI routers
- Storage Array
- Disks



ORACLE

ASM-Database
Character or block device driver
Multipathing driver – PowerPath, Windows MPIO
SCSI driver
HBA – Host Bus Adapter, Qlogic, Emulex
System Bus - PCI
Switch/iSCSI routers – Brocade, McData
Storage Array – EMC, HDS, HP
Disks - Seagate

The database processes makes I/O calls to the block device. The I/O scheduler in Linux forms the interface between the generic block layer and the low level device drivers. The block layer provides functions that are utilized by the file systems, databases (10gR2) and the virtual memory manager to submit I/O requests to block devices. These requests are transformed by the I/O scheduler and made available to the low-level device drivers. The device drivers consume the transformed requests and forward them (by using device specific protocols) to the actual device controllers that perform the I/O operations.



To make sure that a system delivers the IO demand that is required, all system components on the IO path need to be orchestrated to work together.

The weakest link determines the IO throughput.

DW systems require specific considerations due to the high IO data rate

CPU power – ETL processing, Account for non-database I/O processing

Number of HBA – driven by Mbytes/s not IOPS

Server Memory

DW are also mixed workload – transactional + DSS. But size for DSS, since it's the bigger drain.

To increase host IO throughput– add nodes to RAC and distribute workload

Don't oversubscribe on the FC switch

Maybe for OLTP systems:

This SAN fan-out ratio of storage ports typically ranges from 6:1 to 12:1 server-to-storage subsystem ports. This ratio balances different server platforms and applications across these subsystem ports to fully utilize available bandwidth while enabling the maximum throughput of each HBA to achieve near-wire-rate throughput at a given time. The ratio also implies that the Fibre Channel switch ports that the server HBAs are connected to are being underused most of the time. To achieve the best price-performance metrics, it is common practice to oversubscribe server HBA-connected switch ports, as long as storage subsystem ports have access to full line-rate performance and the Fibre Channel switching fabric is non-blocking.

CPU requirements depend on user workload:

Concurrency of users, ratio of CPU-related tasks

Memory requirement mostly user-process driven

IO requirements depend on query-mix:

CPU vs. IO

- Relative CPU power for IO related tasks

Logically Random IOs (predominant in star schema)

- required for index driven queries, e.g. Index lookups, Index driven joins, Index scans

Understand bandwidth limitation and choosing appropriate components

- **Database** – Scales to the ability of the hardware. I/O throughput is bounded by host I/O components; e.g., CPU, SCSI driver, HBA, etc.
  - In DSS environments ensure enough CPU resources are available to accommodate the application - plan for 75 -100MB/sec per GHz/CPU
  - Ensure largest I/O request is defined at the host – this should be 1MB.
  - All database I/O performed against ASM devices are initiated and processed by the database; i.e., ASM is not involved.
- **HBA** – dependent on the type (1, 2, 4 Gbit/s)
  - 2GBit FC HBA – best possible rates:
    - Sequential – 150-180MBytes/s
    - Random rate - is a function of block size, the larger the blocksize the lower the random rate
  - 4GBit FC HBA – ~375 MBytes/s

ORACLE

When configuring hardware for a data warehouse defining the IO requirements is very challenging because predicting the IO demand is difficult.

In general the IO requirements depends on the query mix.

As a rule of thumb assuming today's CPUs (3 GHz Xeon or 2.2 GHz Opteron) an IO bound query can drive about 200MB/s per CPU. Having said this, it depends on the ratio of IO vs. CPU bound queries how high the IO requirements are.

It is also very important to investigate whether the system will have more queries issuing random vs sequential IOs. Random IOs are more dominant in index driven queries such as index lookups, index driven joins such as nested loops or bitmap indexes or index scans. Sequential IOs are more dominant in table scans used in hash joins

Regarding the Random IOPS rate - is a function of block size and the cost of the per block overhead. Max theoretical rate (8KB block) is about 16k IOPS, but a storage system will not respond that fast (100% cache hit rate required).

For NAS (In NFS environments):

```
A single Gigabit Ethernet can support about 30 MBytes/s at a reasonable 30%
utilization or 70 MBytes/s at a high 70% utilization.
```

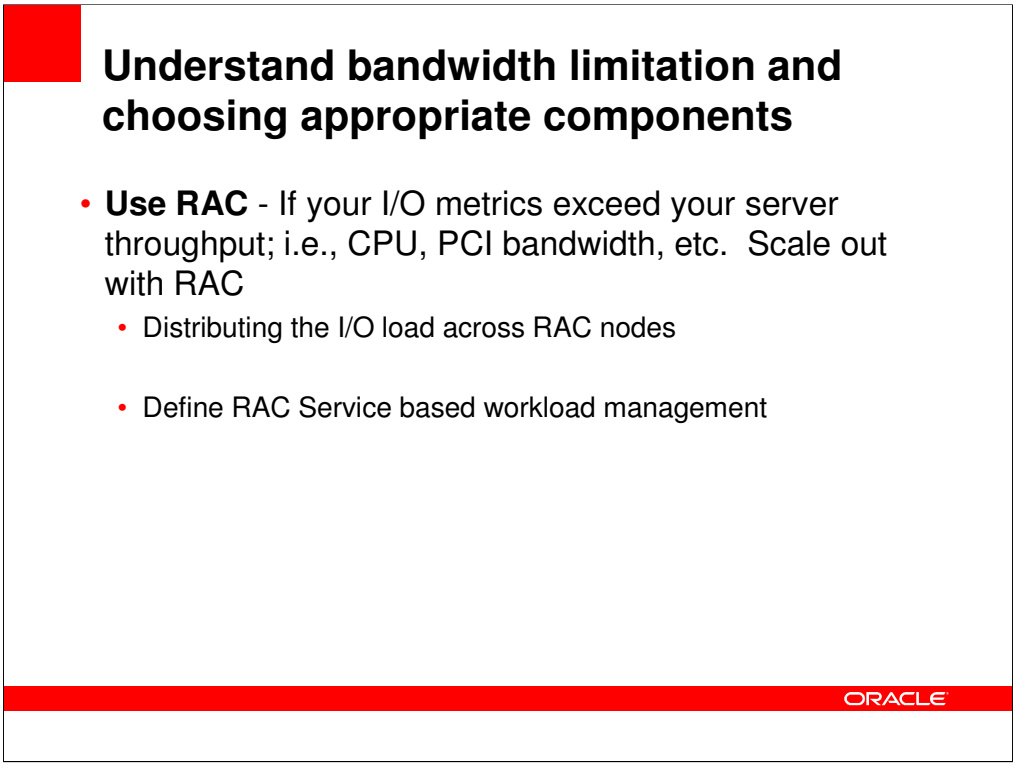
SCSI driver

the per port I/O rates (either random or sequential) are determined by the OS ability to drive I/O and the storage system's ability to respond.

The largest the I/O request from SCSI driver (on Linux 2.6),is 1MByte but observed or typical is 512K. SCSI layer limitation is 512k

SCSI driver limits 128 entries of 4k pages, if two memory locations are next to each other, then these are considered as 1 entry, so you could have more than 512k IO request. Use of large-pages (4MB), PTE 1 entry is 4MB to one physical segment. IA-64 (Itanium) – 16K pagesize, 128 entry * 16K = 2MB.

This part really means look for parts and pieces that match or exceed the IOPS and Mbytes/s requirement, as determined from the previous slide.



PCI Bus:

Account for PCI bandwidth – don't overwhelm PCI bus, by plugging in too many HBAs. The interrupt rate per HBA on that PCI bus is as important as the transfer rates. However, the host side PCI bus bandwidth must be watched, ensure your PCI capabilities are met. PCI-e can do - 2Gb 200Mb/s.

The big concept hers is "lanes". For example, if you want to use 4Gbit FC and run full duplex with a single port HBA, you need 400 MB/sec of bandwidth for each direction. In PCI-E terms, that's two lanes, since each lane will be able to run at full rate. You could use a single lane, but you would be limited to 250 MB/sec. That might not be a problem for IOPS like database index searches.

Using 400 MB/sec transfer with no other overhead and 16 KB requests would support 25,600 (400 MB per second/16 KB requests) requests per second, while 250 MB/sec single lane would support 16,000 requests. There is additional overhead, so you will never really achieve those rates, but either way, one lane or two lanes far exceeds what most servers, HBAs and RAID systems can deliver.

So from an IOPS perspective, a single lane and 4Gb HBA will work just fine, and with dual port, one or two lanes will more than saturate most RAID configurations. Assuming that a disk drive at most can do 150 random I/Os per second, you would need a large number of disk drives or cache hits to run at full rate. Since most RAID controllers do not have a command queue of 8K, you will also far exceed the command queue of RAID controllers.

I can't remember a time when the performance of an I/O bus was faster than the fastest host attachment, so we have reached an important time in technology history where the bus is fast enough to run any card at rate. This assumes a number of things:

Understand bandwidth limitation and choosing appropriate components

- Storage arrays come in different flavors and sizes
  - High-end
  - Modular arrays
  - Low-cost commodity
- What performance number should you look for in a storage array.
  - A published SPC-1 result is the best indicator of how a system will behave with typical DB workloads.
  - Use aggregate I/O metrics to size the storage array
    - describe I/O metrics and application characteristics to Storage vendor.

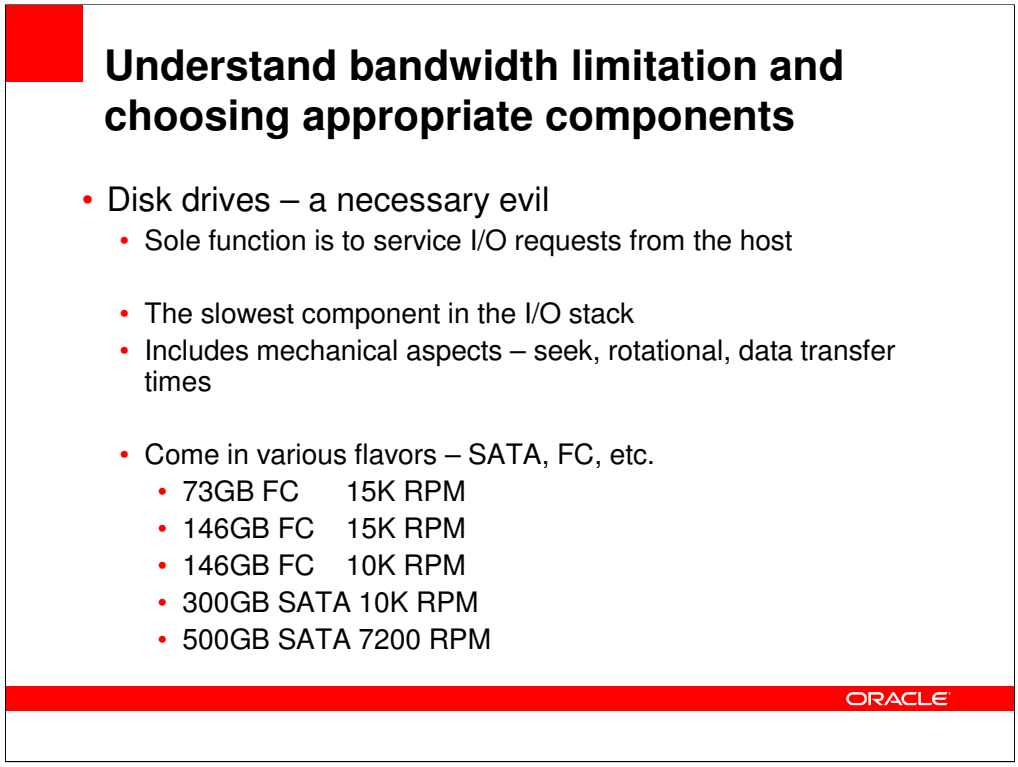
ORACLE

The array architecture is a vast topic, which can be discussed at length. This section addresses the internal bandwidth provided to each disk array path in the enclosure and also to the cache memory.

The disk enclosures or the disks are connected in FC loops. In modular storage these loops can be somewhat customized to provide higher performance. In high-end enterprise arrays, the drive loop technology is already predefined for specific throughput. The internal bandwidth of a storage system for medium performance should be around 4 Gbyte/sec, while for extreme throughput you can choose from 8 Gbyte/sec to 15 Gbyte/sec technology.

The read/write throughput to the cache should also be considered for maximum storage performance. For a high-end array the transfer rate to cache can be as high as 800 Mbytes/s, and for a medium-rated array it could be around 300-400 Mbytes/s.

The big differentiator is the CPU processing speed and onboard bus speed, this dictates how fast (well) IO requests are passed down through to the back-end adapters and disks.



Determining disk IO Throughput (IOPS):

1. Divide 10000 RPM by 60 seconds (10000/60 = 166 RPS)
2. Convert 1 of 166 to decimal (1/166 = 0.0006 seconds per Rotation)
3. Multiply the seconds per rotation by 1000 milliseconds (6 MS per rotation)
4. Divide the total in half (6/2 = 3 MS) or Rotational Delay
5. Add an average of 3 MS for seek time (4 MS + 3 MS = 7 MS)
6. Add 2 MS for latency (internal transfer) (7 MS + 2 MS = 9MS)
7. Divide 1000 MS by 8MS per IO (1000/9 = 111 IOPS)

Note this is the effective IOPS that can be gotten from a 10k RPM drive. Generally I reduce this value by 10% to account for seek overhead

Each time an application issues an IO, it takes an average of 9MS to service that IO on a 10K RPM disk. Since this is a

fixed time, it is imperative that the disk be as efficient as possible with the time it will spend reading and writing to the disk.

The amount of IO requests are often measured in IOs Per Second (IOPS). The 10K RPM disk has the ability to push 80 -

to 100 (burst) IOPS. To measure the effectiveness of IOPS, divide the amount of IOPS by the amount of data read or

written for each IO.

Which disk type is right for me?

- No rigid rules on which disks should be used
  - Consider cost and performance
- General guidelines:
  - SATA disks –
    - Are very good for sequential I/O – archived logs, flashback logs, RMAN backups.
    - For 2nd, 3rd tier of tiered storage architectures (ILM, HSM)
    - In Low cost storage solutions
  - High-speed FC disks great for heavy IOPS and high sequential throughput applications.

ORACLE

Understand bandwidth limitation and choosing appropriate components

Drive Type	RPM	IOPS @ 9ms (95%tile)	Sustained throughput
FCAL	10,000	80	20-30 MB/sec
FCAL	15,000	110	25-35 MB/sec
SATA	7,200	50	20-30 MB/sec

A word on disk bandwidth –

IOPS and Throughput are mutually exclusive – you won't get 25MBytes/s &110 IOPS at the same time.

ORACLE

Becareful of IOPS and Mbytes/s that come from the disk vendor, these numbers are for small IO requests (512bytes).

Typically you should assume a 20-30% reduction from this vendor value (this accounts for the disk seek, positioning overhead).

Understand bandwidth limitation and choosing appropriate components

- So how many physical disks do I need to drive my App?
  - Use your IOPS or Mbytes/s metric to determine the number of physical disks.
  - Assume no storage array I/O cache hit ratio in the calculation
- Two examples –
  - Requirement is 600MBytes/s:
    (600MBytes/s) / (25MBytes/s per disk) = 24 disks
  - Requirement is 3000 IOPS:
    (3000 IOPS) / (110 IOPS per disk) = 28 disks
- Note that this raw disk; i.e; not mirrored or RAID'ed

ORACLE

Disk Configuration - Best Practices

- Use large number of disks in the backend
  - Keep it simple – no need to separate out random I/O from sequential
  - Its been proven that greater number of disk spindles can consume any I/O workload type.
- Ensure disks span multiple backend disk adapters
  - On most high-end enterprise arrays this is automatic
  - Midrange arrays may require manual configuration
- Implement multiple access paths to the storage array using two or more HBAs or initiators
  - Use multipathing for load balancing and failover

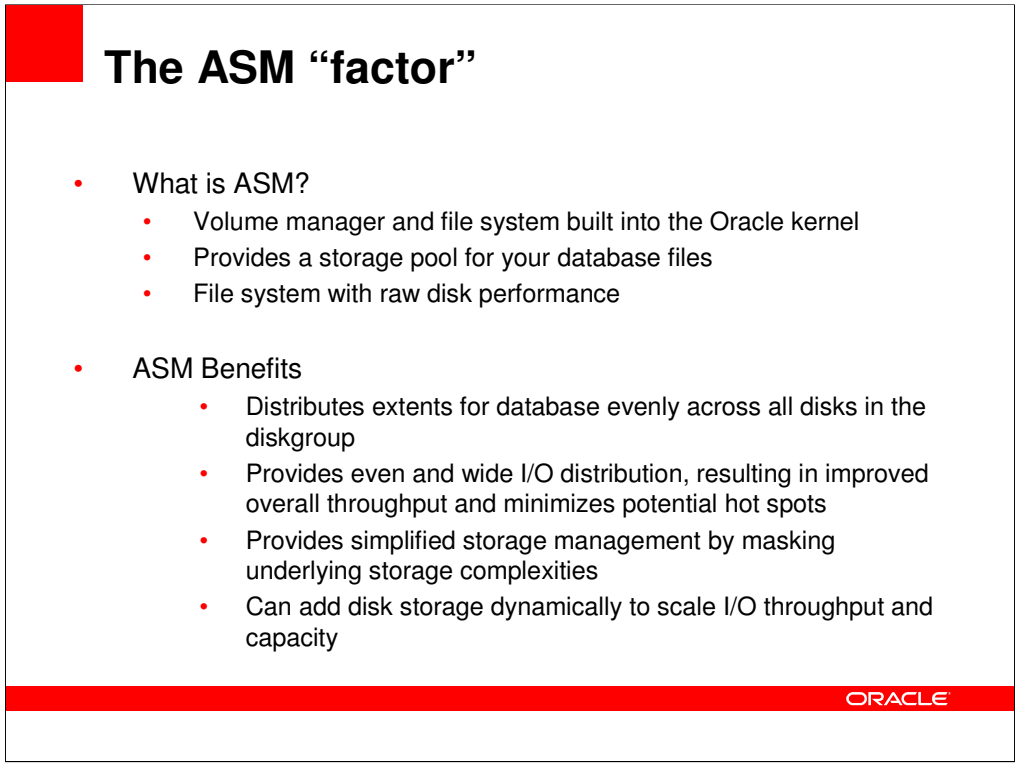
ORACLE

•Make sure disks span multiple backend disk loops on midrange systems (automatic in enterprise)

•Carving too many LUNs same the RAID group will not give you any extra IOPS/Mbytes,as the RAID group has finite bandwidth

•Multiple HBAs provide multiple access paths to a the diskgroup disks. Multi-pathing software includes PowerPath, MPxIO, or Secure Path.

•Do the same for just fail-over purposes for midrange systems (all midrange systems have controller-based ownership of LUNs or RAID Groups – it is very costly to constantly dynamically change this in the storage controllers).



The following are some key benefits of ASM:

- I/O is spread evenly across all available disk drives to prevent hot spots and maximize performance.
- ASM eliminates the need for over provisioning and maximizes storage resource utilization facilitating database consolidation.
- Inherent large file support.
- Performs automatic online redistribution after the incremental addition or removal of storage capacity.
- Maintains redundant copies of data to provide high availability, or leverage 3rd party RAID functionality.
- Supports Oracle Database 10g as well as Oracle Real Application Clusters (RAC).
- Capable of leveraging 3rd party multipathing technologies.
- For simplicity and easier migration to ASM, an Oracle Database 10g Release 2 database can contain ASM and non-ASM files. Any new files can be created as ASM files whilst existing files can also be migrated to ASM.
- RMAN commands enable non-ASM managed files to be relocated to an ASM disk group.
- Oracle Database 10g Enterprise Manager can be used to manage ASM disk and file management activities.
- ASM reduces Oracle Database 10g cost and complexity without compromising performance or availability.

The ASM “factor”

- A walk through configuration of a diskgroup
  - Present LUNs to host
  - Ensure correct disk permission so that ASM disk discovery will “find” the provisioned LUNs.
  - Create diskgroup using the required number of disks:

```
SQL> create diskgroup DATA external redundancy
disks ‘/dev/sda1’, ‘/dev/sdb1’, /dev/sdc1’,
‘/dev/sdd1’;
```

  - Create you database files in the ASM diskgroup

```
SQL> create tablespace OOW_TBS datafile ‘+DATA’
   size 200G;
```

ORACLE

ASM Diskgroup Best Practices

- Create two diskgroups. One for database area and another for recovery area
- Create diskgroups using large number of similar type disks
  - same size characteristics
  - same performance characteristics
- Typical question: *How many disks do I need in my ASM diskgroup and what should be the LUN size?*
  - 1. How disks ? We answered it! Its driven by the I/O requirements
  - 2. LUN size is dependent on site-specific storage standards. LUN size is not relevant as much as the number of LUNs.

ORACLE

These disks should of similar size and performance characteristics.

In DSS environments it maybe prudent to create a third diskgroup for Tempfile; e.g;, a TEMP_DG.

RAID 10 or RAID 5 storage array LUNs can be used as ASM disks, to minimize the number of LUNS presented to the OS.

; examples would EMC Metavolumes or HDS Sfvols.

Piecing It All Together
A Customer example

Customer Overview

- Customer application is a DSS system
  - Current size 1.1TB
  - I/O rate is 133MBytes/s and 2247 IOPS
- Customer wants to consolidate two more databases, roll in more users as per business requirements.
  - I/O requirements will grow to 800 Mbytes/s
  - Database will grow to 2TB
  - Will be implementing RAC to support various services workloads

ORACLE

An ‘average’ DW system should plan for 75 -100MB/sec per GHz/CPU

Typical mixture of IO and CPU intensive operations

Ball park number, adjust accordingly

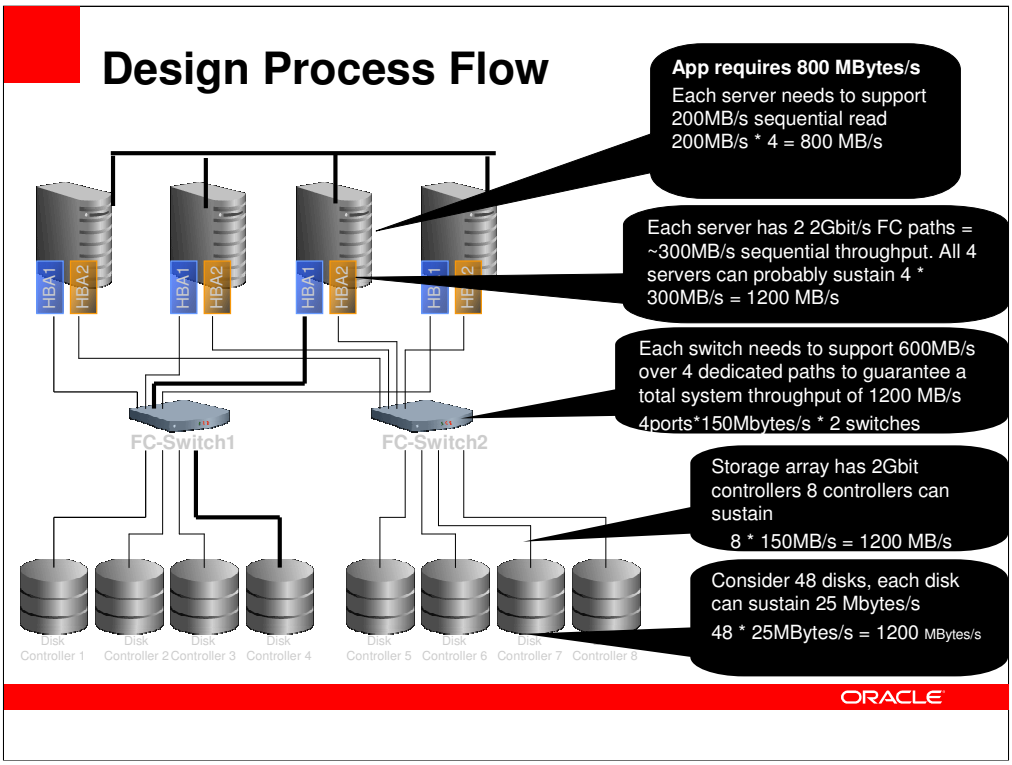
Customer AWR report

"Instance Activity Stats" Per second stats

physical read total IO requests	1,197.43
physical read total bytes	67,932,081.00
physical write total IO requests	1,050.86
physical write total bytes	65,114,880.37

Total:
physical read total I/O requests + physical write total IO requests ~ 2247 IOPS
physical read total bytes + physical write total bytes ~ 133 Mbytes/s

ORACLE



To make sure that a system delivers the IO demand that is required, all system components on the IO path need to be orchestrated to work together.

The weakest link determines the IO throughput.

In this example -

On the left side you see a high level picture of an system. This is a system with 4 nodes, 2 HBAs per node two fibre channel switches, which are attached to 4 disk arrays each. The components on the IO path are the HBAs, cable, switches and disk arrays. Performance depends on the number and speed of the HBAs, switch speed, controller quantity and speed plus number and speed of disks. If any of these components are under configured, the system throughput is determined by this component.

Assuming we have 2GBit HBA, the nodes can read about 8 times 200MB/s = 1.6GBytes/s. On the other hand assuming each disk array has one controller, all 8 arrays can also do 8 times 200MB/s = 1.6GBytes/s. Hence, each of the fibre channel switches also need to deliver at least 2GBit/s per port, to a total of 800 MB/s total throughput. The two switches will then deliver the needed 1.6 Gbytes/s

When sizing a system also take the system limits into consideration. For instance, the number of Bus Slots per node is limited and may need to be shared between HBAs and Network cards. In some cases dual port cards exist if the number of slots is exhausted. The number of HBAs per node determines the maximal number of fibre channel switches. And the total number of ports on a switch limits the number of HBAs and disk controllers.

Customer Example – implemented configuration

- RAC 10gR2 - 4 servers, each server has 8 CPUs, with 2 HBAs
- EMC Clariion CX3-40
- 4 DAE trays of 15 drives each - 146GB 15k rpm FC drives
- (13) 260GByte RAID 1/0 LUNs (52 spindles backend) were used to create 13 ASM disks for DATA diskgroup.

ORACLE

CX3-40, 4GB memory per SP.

4 DAE trays of 15 drives each. 146KB 15k rpm FC drives.

Base (tray 0) includes the first 5 drives reserved for EMC array software and vault. Drives 13/14 on that DAE enclosure (enclosure 0) configured for hot spare.

The drives in enclosure 1,2, and 3 are grouped together to form 2+2R1/0 LUNs. 13 such LUNs (52 spindles) were used to create 13 ASM disks, each with usable capacity of 260GB each. Total usable capacity was 13 x 260 = 3.3GB. Two additional SATA trays were also configured to be used for FRA with 7200 rpm SATA drives.

Customer Example – implemented configuration

- New Production AWR report – node1

```
physical read total IO requests          1,797.23
physical read total bytes           89,232,081.00
physical write total IO requests         1,550.44
physical write total bytes          75,227,212.40
```

Total:

physical read total IO requests + physical write total IO requests = 3347 IOPS

physical read total bytes + physical write total bytes ~ 165 Mbytes/s

*This report needs to be aggregated for all 4 nodes in the cluster.
Aggregated Mbytes/s across 4 nodes ~ 660Mbytes/s

ORACLE

Customer Example – implemented configuration

- Let's see how well their IO behaves:

Top 5 Timed Events

Event	Waits	Time (s)	Avg wait (ms)	%Total Call Time	Wait Class
CPU time		173,911		32.2	
PX Deq Credit: send blkd	9,462,161	124,532	13	23.1	Other
db file scattered read	3,834,332	56,639	15	10.5	User I/O
direct path read	3,604,017	32,753	9	6.1	User I/O
latch: cache buffers chains	16,494,723	18,563	1	3.4	Concurrenc

ORACLE

The "PX Deq Credit: send blkd" wait event is an idle wait event. This wait occurs in PQO, when a process wishes to send a message and does not have the flow control credit. Process must first dequeue a message to obtain the credit. Indicates that the receiver has not dequeued and/or completely consumed the prior message yet.

Validating the configuration

- Validate the configuration in a test environment
- Inject representative workload when possible
  - Use 11g Real Application Testing to replay current workload.
- Use Swingbench or Orion to mimic the production IO workload characteristics

ORACLE

Verify that the IOPS/MBytes will be able to be sustained in the proposed configuration.

Use ORION to validate (a calibration tool) that IOPS/MBytes will be able to be sustained in the proposed configuration.

Summary - The 4 take-ways

- Determine Application SLA and I/O requirements
- Understand each I/O component's bandwidth limits
- Design for scalability, availability, and throughput

For More Information

search.oracle.com

or

otn.oracle.com/asm

ORACLE


ORACLE®