Tune your Oracle Automatic Storage Management

Tips & Techniques at implementing and optimizing your database storage using Oracle ASM in a SAN environment

by Thiru Vadivelu

Implementing Oracle's Automatic Storage Management in a Storage Area Network environment can be an interesting and involved task. The purpose of this document is to provide some generic guidelines and things to consider/configure when setting up/migrating to Oracle Automatic Storage Management (ASM), in a SAN environment, based on our past experiences. This article is primarily intended for DBAs, Database Managers, System administrators and Storage administrators.

Why ASM?

- ASM is an Oracle provided Database/Cluster filesystem and Storage Volume manager, optimized for Oracle for any kind of workload (be it OLTP or DSS or Batch, sequential or random access), built-in with the database, with no additional cost.

- It follows SAME (Stripe and Mirror Everything) strategy, eliminating any database level IO hotspots, as the database files are striped equally across all the available disks in the allocated disk groups and are dynamically rebalanced with the addition and removal of disks online.

- ASM provides for better storage utilization and storage consolidation (ie allows multiple databases on the same Server or across multiple Servers to share storage through Oracle clusterware).

- It allows leveraging existing storage RAID technologies (Mirroring, RAID 5 etc) to provide for additional striping and redundancy.

- It can leverage IO multipathing technologies (like EMC Powerpath, Hitachi HDLM etc) to provide for better IO balancing and fault-tolerance.

- Makes use of Oracle 'Kernel Service Direct File (ksfd)' interface to perform KAIO (kernel asynchronous IO) against raw devices by-passing OS page cache, providing the best IO performance with no overhead from Operating system or conventional Logical Volume Managers.

What ASM is not?

- Although, it's a 'database filesystem', it does not provide storage for Oracle binaries, certain configuration files, trace files, logs etc.
- It cannot resolve database data block contention at the segment level.

- It cannot automatically improve IO performance when the SAN disks are experiencing high 'disk service times' possibly contributing to high IO wait times at the Oracle layer.
- It does not provide IO Multipathing capabilities of its own.
- It does not provide any memory caching/buffering, by itself (like OS or SAN frames).Only the ASM metadata is cached in the ASM instance.

Database Storage Planning/Strategies:

In general, the following are the primary factors, when it comes to planning database storage:

- IO throughput (and I/O per sec)
- IO service/wait times
- Volume of storage
- I/O workload (ie nature of access like sequential r/w , random r/w)
- Availability

To achieve the desired level of the above parameters which are based on application requirements, the following things need to be strategized and tuned:

- Storage Tier level
- Raid group/configuration
- Size/Number of Luns(Logical Unit Number)
- Storage Frame Caching/Sharing
- SAN Stripe size
- IO Multipathing
- Bandwidth of IO path
- Volume/Filesystem configuration
- OS Page/Buffer Cache
- ASM diskgroups configuration
- ASM Instance configuration
- ASM allocation unit size
- ASM fine grained stripe size
- ASM Max IO size
- Oracle Block size
- Oracle Multiblock read count
- Oracle Caching(SGA/PGA)

It is not the purpose of this document to go into details of all the above requirements and technologies (purely based on application needs and technology availability), but to merely share my experiences and provide some guidelines, with respect to Storage/ASM setup.

Storage Tier level:

- Greatly dependent on budget, availability, and storage team's standards and application's desired IO response time.
- For production applications, consider either Tier-2 or Tier-1 storage and for development Tier- 3.
- For Multiblock reads(common in Full table scans and Index fast full scans), Tier-1 storage should provide an average IO wait time in the range of 10-15ms(with Tier-2 in the range of 20-30ms) , sufficient enough to achieve an overall IO response time requirement in this range.
- For Single block reads (common in Indexed reads), Tier-1 storage should provide an average IO wait time in the range of 1-5ms(with Tier-2 in the range of 5-10ms).
- In some of our benchmarks, We didnt realize significant response time difference between Tier-1 and Tier-2 , when

    i) Most of the IO is being satisfied by SAN frame cache and did not stress the underlying San storage enough.

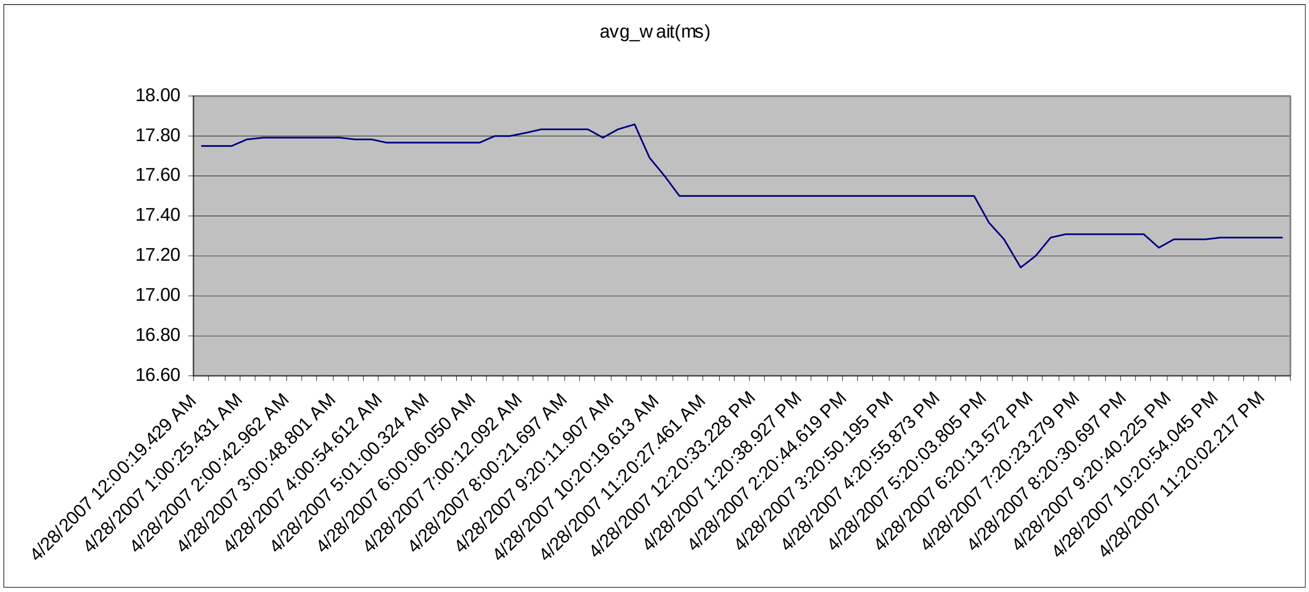
    ii) The DB Server's cpu utilization is very high, because of heavy amount of Logical IO .In this case Oracle was not able to make sufficient IO requests to force high IO throughput with the Tier-1 storage.

Its not to be taken that Tier-1 will not significantly perform better than Tier-2 (after all it provides for better striping across more luns), but the benefit is dependent on other factors such as attainable IO throughput, San caching, Raid configuration etc.

For eg, in one of our benchmarks (conducted in a Datawarehouse environment dominated by large sequential reads with 95% of IO satisfied by SAN cache), the average IO wait was almost the same between Tier-1 and Tier-2

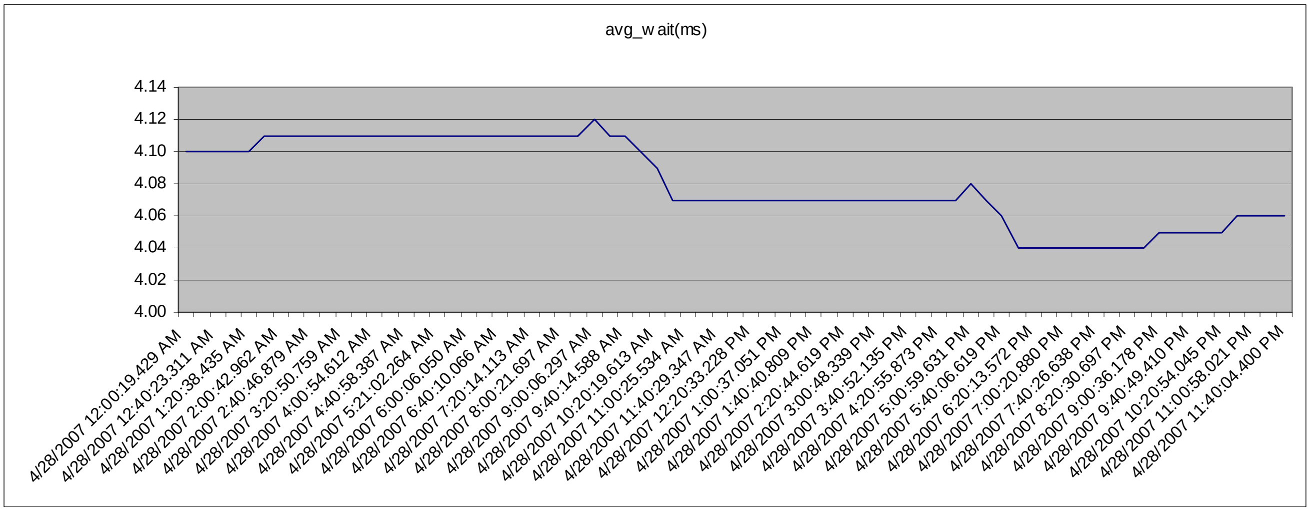
Average IO Wait(ms) over the Tier2 to Tier1 migration: (Full table scan)

the Avg IO Wait(ms) almost remained the same between the two(from about 17.8ms in Tier2 to about 17.2-17.4 ms in Tier1) . The improvement was very minor.



AvgIO_Wait(ms) over Tier2 to Tier1 migration (Indexed Reads):

the Avg IO Wait(ms) almost remained the same between the two (about 4.10-4.12ms with Tier2 and 4.06-4.08ms with Tier1).The improvement was very minor, again..



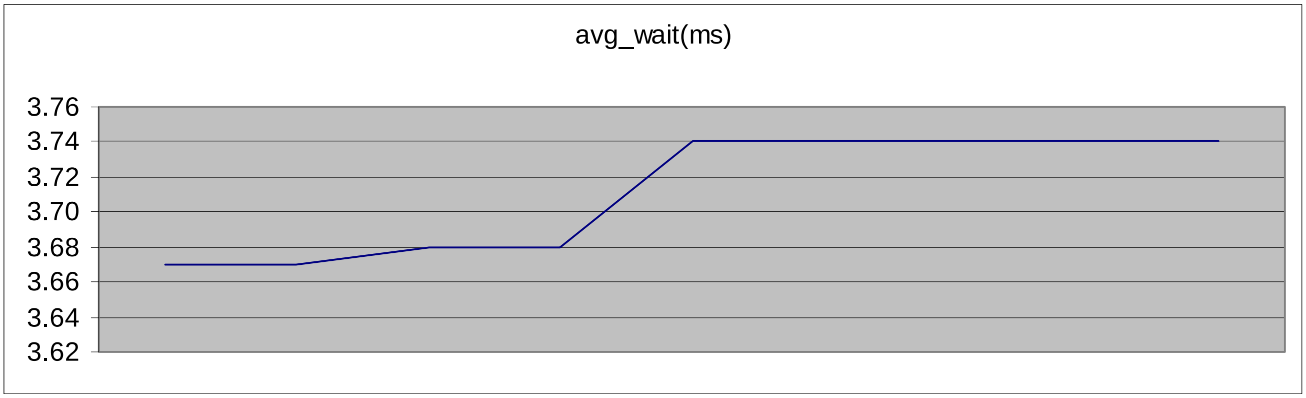
Also the graphs plotting ‘SQL Response Time’ and ‘DBTime/sec’ did not show significant difference between Tier-1 and Tier-2.

Raid group/Configuration

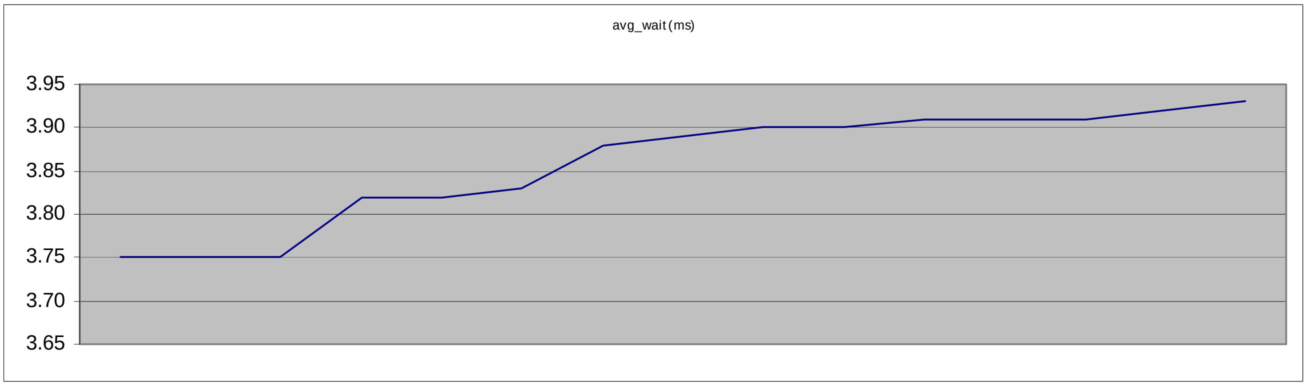
- RAID 1+0(or 0+1) provides the highest performance and availability because of striping and mirroring, but also the most expensive configuration. It’s a viable option for performance intensive applications which require high availability.
- RAID 1+0 is marginally better than RAID 0+1 for availability (for multiple disk failures), since the individual disks are mirrored first and then striped as opposed to 0+1, where all the disks in the volume are striped first and then mirrored as a whole.
- Raid 1(Mirroring) , when used with ASM, behaves functionally equivalent to Raid 0+1, since ASM performs file extent striping first and then the Luns are mirrored at the SAN layer.
- In one of the benchmarks we did in a Datawarehouse environment, RAID 5 (Tier1) outperformed RAID 1(Mirroring /Tier1) marginally (5%) and this can be directly attributable to ‘double-striping’ (ASM+ Raid 5 striping), but keep in mind of the availability benefits with Mirroring.
- RAID 5 is a very viable option for Datawarehouse applications which do predominantly ‘large sequential reads’ and ‘bulk writes’, as opposed to OLTP applications which are dominated by random read/writes. The performance penalty experienced in this configuration due to the need to perform 4 IOs at the Raid level for every database IO (to cater to parity read/recalculate/write operations) is greatly minimized by the presence of a huge San frame cache. The Database IO is asynchronous (ie doesn’t wait for acknowledgement from the disks) and the reads/writes are buffered at the San cache.

Raid 1 vs Raid 5: Benchmark done on a Datawarehouse Database

Raid 5 (Avg IO Wait(ms) for Indexed Reads)



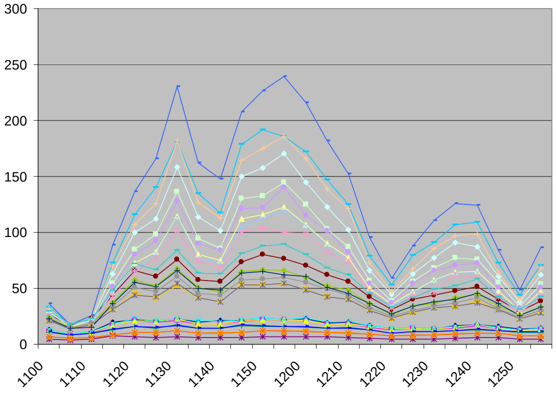
Raid 1 (Avg IO Wait (ms) for Indexed Reads)



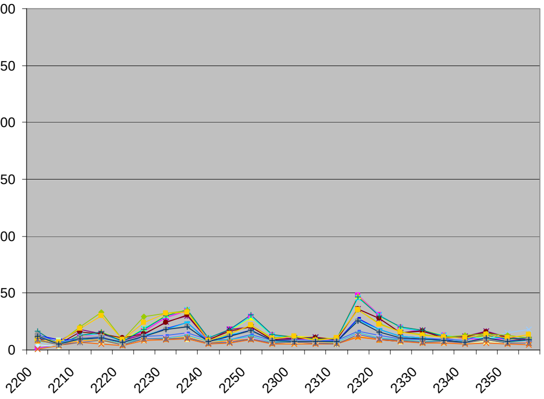
- Also in these benchmarks, we determined that sharing the same raid-groups across multiple LUNS results in IO contention, which we were able to resolve by allocating the disks from One Raid-group to One LUN.

The following benchmark results compares the average IO response time of busiest luns between the 2 configurations i) 16 Luns in 2 RAID groups ii) 16 Luns in 8 Raid groups

Case i)



Case ii)



Size/Number of Luns:

- Although Oracle’s recommendation is to minimize the number of LUNS per diskgroup by having fewer larger Luns, to minimize the LUN Management overhead, our experiences show that IO performance is more dependent on the number of underlying disks, the LUNS are spread across. Nevertheless, its worth testing fewer/larger luns spread across as many physical disks as the configuration allows, based on the expected IO throughput and service times.
- Our benchmarks show that, for the same Tier configuration, spreading the diskgroup across double the luns(double the number of disks), improved performance significantly.
- Since the ASM file extent distribution strategy is capacity based, its strongly recommended to size all the luns in a diskgroup, the same (and have same characteristics).Otherwise Lun contention will result as the larger luns will have more ASM file extents stored.

SAN Stripe size:

- Ideally, the RAID stripe size at the SAN layer should match ASM stripe size (1MB by default).
- If the above is not possible, since the San frame is commonly shared across multiple applications, then a stripe size of 256K or 512k should be ok.

I/O Multipathing:

- ASM can make use of various Multipathing technologies such as EMC Powerpath, Hitachi HDLM, Sun Powerpath etc) and is highly recommended to provide IO path redundancy, IO load balancing.
- For eg ,on AIX, ASM can directly access the multi-pathed raw lun devices(/dev/rhdiskpower*)

OS Buffer/Page Cache:

- Since ASM uses the raw devices directly and performs a DIRECT KAIO, the OS page cache is by-passed and hence need to be minimized (ie resized down) when migrating from a buffered filesystem to ASM.
- For eg, on AIX, the page cache kernel parameters ‘minperm’ and ‘maxperm’ default to 20 and 80 respectively which can allocate quite an amount of OS memory, which is redundant and actually detrimental to performance since this memory is not available for Oracle SGA/PGA. Recommended to resize these parameters to say 2 and 20 respectively to allocate minimal page cache and use that, instead for Oracle’s SGA/PGA.

ASM diskgroups:

- To achieve maximum storage utilization and consolidation, not more than 2 diskgroups (ie one for Data/Index/Undo/Redo/Temp and one for FLASHBACK (flashback logs,1 copy of online redolog,1 copy of controlfile, archivelogs)) is generally required, for all the database storage.

- When the underlying SAN storage is providing the redundancy, configure ASM with 'External Redundancy' with 1 Failure group.
- Normal Redundancy (default: 2 failure groups) and High Redundancy (3 failure groups) ASM configurations are options, when configuring the same diskgroup across multiple storage frames or across datacenters for redundancy.

ASM Clustering:

- Oracle's Clusterware (primarily consisting of CSS and CRS daemons) gets installed by default when ASM is installed, although you will find only the CSS process running in single instance configurations. This allows for the ASM instance to register its diskgroups, Oracle_Home, Oracle_Sid with the CSS service, used by the database instance to communicate with the ASM instance.   Oracle's Clusterware is offered at no additional license cost.
- To achieve maximum storage utilization and consolidation, it's recommended to share the same ASM diskgroups across multiple databases residing on the same server (although it could result in single point-of-failures, with respect to diskgroup). This is a very viable option when it comes to development and test databases and should be carefully considered for Production databases.
- It is also possible to share the same ASM diskgroups across different databases running on multiple servers by clustering ASM instances (through Clusterware/RAC). However, this would require RAC license for clustering ASM instances. The cost and complexity of this configuration may not do enough justice to warrant this configuration.

ASM allocation Unit:

- Determines the unit of ASM storage (ie size of file extent , also called as stripe depth) and defaults to 1M.This is sufficient for most installations.
- However for VLDBs(multi-terabyte and greater), its recommended to increase the size of the allocation unit(undocumented parameter "_asm_ausize") from 1M to say 8 or 16M, to minimize the ASM overhead in opening the files and caching the metadata.. This can only be done at the time of diskgroup creation and cannot be altered for an existing diskgroup.

- Benchmarks from HP show a 16% performance improvement when the allocation_unit size was increased from 1M to 8M.

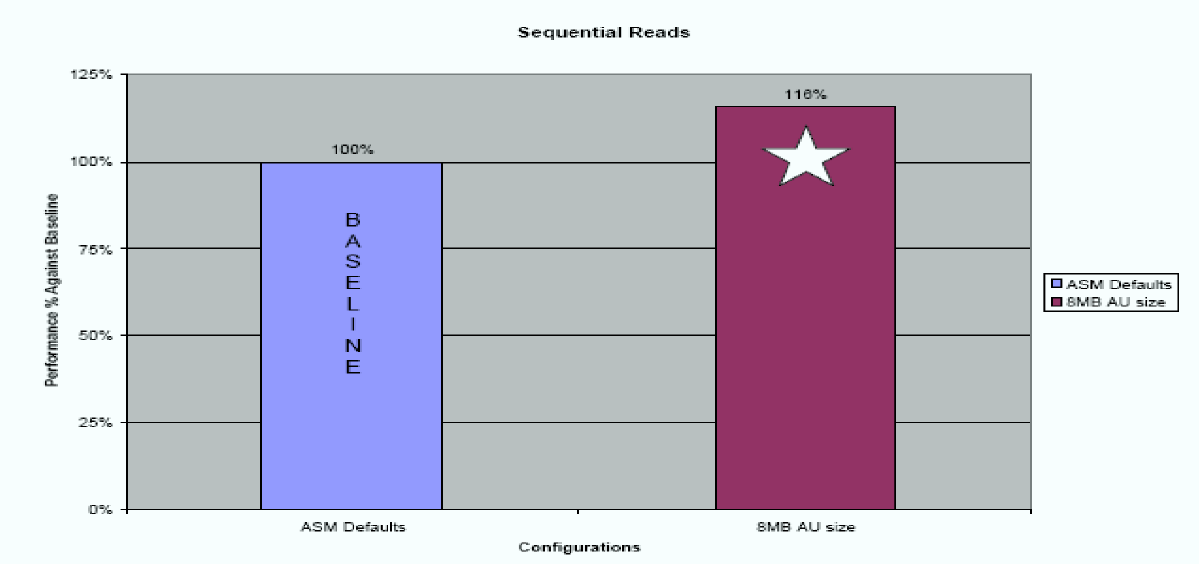


Figure 3 - DSS Results

In this test, the DSS scripts completed much faster when the ASM allocation unit size was set to 8MB, as shown by the right-most column (it outperformed the baseline by completing in less time). The sequential reads appear to function better when the ASM allocation unit is set to 8MB. In this test, setting the ASM AU size improved overall DSS query performance by 16%.

ASM Stripe size:

- Oracle provides COARSE striping for big IO involving Datafiles and FINE striping for small IO involving online redologfiles/controlfiles. The coarse stripe size is 1M and Fine striping size is 128K. These are optimal sizes for most installations
- As outlined earlier, increasing the allocation_unit (coarse stripe depth) size from 1M to higher values are beneficial only for VLDBs and should be set at the instance level , when creating the diskgroups.
- The FINE stripe size is controlled by an undocumented parameter "_asm_stripesize " which is adequate for most installations and increasing this to higher values(say 1M), should only be considered for VLDBs ,following benchmarks.

Oracle Db_FileMultiblockReadcount:

- Should be set such that DB_BLOCK_SIZE * DB_FILE_MULTIBLOCK_READ_COUNT= ASM Coarse Stripe Size = MAX_IO size = 1M(default)

ASM Max IO:

- The maximum size of an individual IO request is determined by an undocumented parameter “_asm_maxio” which defaults to 1M, which is sufficient for most installations.
- My attempts to increase this to 2M , did not result in increasing the maximum_io, inspite of increasing the db_file_multiblock_read_count to 500, since its most probably limited by the Oracle internal hard-coded parameter ‘SSTIOMAX’ , defaulted by Oracle to 1M on most platforms , at the installation time.

ASM Installation:

- ASM software (and the Clusterware) is included in the 10G installation media and provided at no additional cost.
- Strongly recommended to create only one ASM Oracle_Home/Instance on a given Server and let all the database instances (both Single Instance and RAC instance) share the same ASM instance, on that node.
- Clusterware gets automatically installed when installing ASM and the CSS (Cluster Synchronization services) daemon starts up on System startup.
- In a RAC installation, each node will have an ASM instance (clustered) which will all mount the same database diskgroups.

ASM Instance configuration:

- ASM instance is used to cache ASM metadata (diskgroups/disks/file extent information etc) only and doesn’t require more than 100MB typically.
- Strongly recommend to use spfile, instead of client parameter pfile. This allows oracle to automatically update the configuration file when new diskgroups are created/mounted.
- Asm_diskgroups parameter identifies the diskgroups to be automatically mounted on instance startup.eg) *.asm_diskgroups='DB_DATA','DB_FLASH'
- Asm_diskstring parameter identifies the OS search path ASM uses to search for disks when discovering disks/configuring diskgroups. Eg) *.asm_diskstring='/dev/rasm*'   . ASM Oracle_Home owner should have read/write access on the paths identified by this parameter.
- Asm_Powerlimit parameter identifies the number of ASM Rebalancing slaves employed to perform diskgroup rebalance operations. Set it to 11, for the fastest (but most IO intensive) rebalancing operation. Can be overridden at the statement level, when reconfiguring disk groups.
- Set Instance_Type=’asm’ to indicate that this is an ASM instance and not a regular ‘RDBMS’ instance.
- For Single instance configurations, Set Instance_name=’+ASM’ (the default) and for RAC,instance_name=+ASMn ,where n=1,2,3…n .

ASM Instance connectivity:

- Since ASM instance does not mount a database of its own or have data dictionary, access to ASM instance is provided normally by OS Authentication (‘/ as sysdba’).

- Recommended to configure Password/SYSDBA authentication via ASM listener for remote logins.

Oracle continues to enhance ASM, introducing new features and functionality with every new Oracle release. Take a look at the new ASM features introduced in 11G, for instance → http://www.oracle.com/technology/products/database/oracle11g/pdf/automatic-storage-management-11g-new-features-overview.pdf

ASM is one the best things that has happened to Oracle since a long time and is quickly maturing into the de facto standard for Oracle database storage. It is the most cost-effective automated database storage solution for Oracle and Grid computing as it eliminates the need to perform guesswork when implementing and tuning database storage, eliminates the need for costly LVMs, eliminates the need to perform manual IO rebalancing, eliminates unnecessary downtime with storage reorganization and vastly improves DBA's productivity. It allows the company's storage to be most effectively used and consolidated leveraging Oracle's cluster technology.
It is the present and future.

Thiru Vadivelu(tmgn12@gmail.com) manages the 'Performance & Capacity Strategies' group in the Corporate technology Risk&Business Services division of JP Morgan Chase,Delaware,USA.He is an Oracle Certified Master (http://www.oracle.com/technology/ocm/tvadivelu.html) and specializes in Performance tuning and High availability solutions involving Oracle technologies.