An Oracle White Paper
November 2013

Technical Comparison of Oracle Database 12*c* vs. Microsoft SQL Server 2012

Focus on High Availability

Executive Overview

Today’s businesses depend heavily on their databases. Should applications and data become unavailable, the entire business may halt. Revenue and customers may be lost and penalties may be incurred. Bad press can have a lasting effect on both customers and stock prices. Certainly, providing continuous data availability is essential for today’s businesses.

Oracle Database 12c comes with an integrated set of High Availability (HA) capabilities that help organizations ensure business continuity by minimizing the various kinds of downtime that can affect their businesses. These capabilities take care of most scenarios that might lead to data unavailability, such as system failures, data failures, disasters, human errors, system maintenance operations and data maintenance operations. As the rest of this document shows, Microsoft SQL Server 2012 database provides functionality for both high availability and data protection which are still multiple releases behind Oracle in terms of the breadth and depth of HA functionality.

Oracle is the database that runs mission critical, highly available enterprise applications for well-known global companies such as NTT DOCOMO, CERN, VocaLink, Turkcell, National Australia Bank, HDFC Bank, Coca Cola, Thomson Reuters, MetLife, Starbucks, Sabre, Monsanto and Telefonica. When it comes to the ability to provide reliable, highly available and continuous service to customers, Oracle is the database of choice over competing solutions such as Microsoft SQL Server.

.

Introduction

Any organization evaluating a database solution for enterprise data must also evaluate the High Availability (HA) capabilities of the database. Data is one of the most critical business assets of an organization. If this data is not available and/or not protected, companies may stand to lose millions of dollars in business downtime as well as negative publicity. Building a highly available data infrastructure is critical to the success of all organizations in today's fast moving economy.

This document provides an in-depth comparative assessment of the HA capabilities available with Oracle Database 12c, and Microsoft SQL Server 2012. The intended audience of this document are IT managers, architects and executives who are evaluating these two databases for their businesses, and are interested in knowing to what extent the HA capabilities of these databases can protect their data and maintain business continuity.

Unplanned and Planned Downtime

One challenge in designing a high availability IT infrastructure is examining and addressing all possible causes of downtime. Downtime can be classified into two primary categories: unplanned and planned. IT organizations should consider potential causes of both unplanned and planned downtime while designing a fault tolerant and resilient IT infrastructure.

Unplanned downtime primarily results from system failures or data failures (e.g. because of human errors, disasters, data corruptions). While such failures may be infrequent, the magnitude of their adverse impact on business operations is significant, leading to high costs of downtime. Planned downtime, on the other hand, is caused by scheduled maintenance activities (e.g. data changes, system upgrades), which are part of the daily operations of any data center. The challenge is to complete the maintenance activity as transparently as possible causing minimal to no disruptions to business operations.

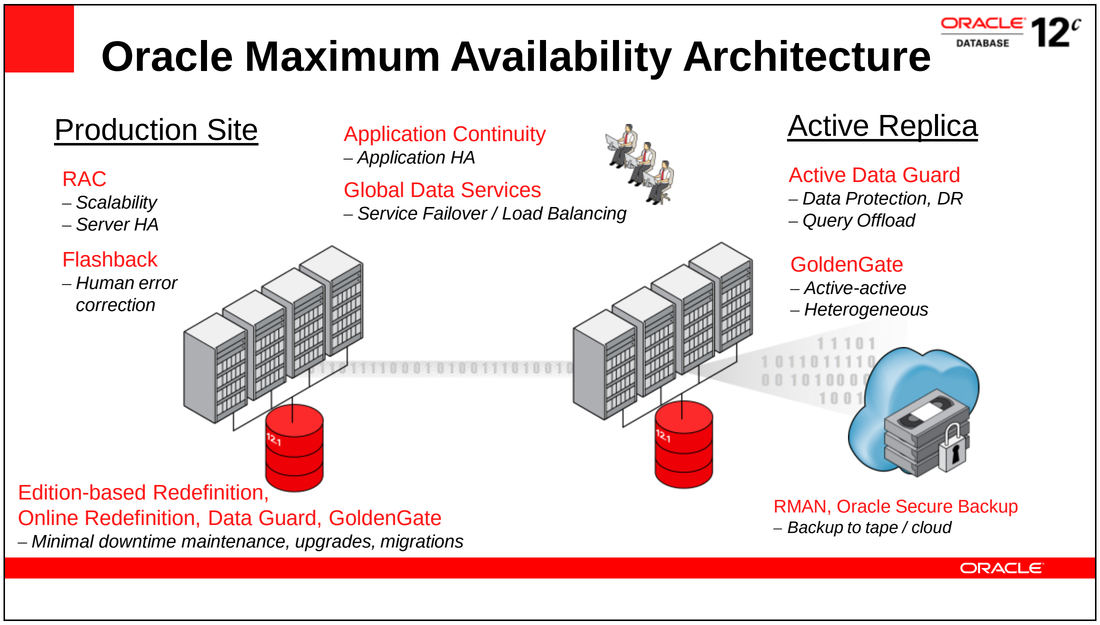
IT managers interested in a high availability solution to meet the demands of their businesses must assess these solutions based on some key metrics, such as:

- The comprehensiveness of their HA capabilities to address various causes of downtime
- Ease-of-use to manage and adapt these solutions to changing business requirements
- Ability to utilize redundant components for effective business use and maximum return on investment.

Any solution that is comprised of a disjointed set of technologies that address HA issues in an isolated manner but not in an integrated fashion, and/or a solution comprised of a lot of redundant, but basically idle components, will not meet the demanding HA requirements of today’s enterprises. With this perspective, the remaining sections perform an HA-based analysis of the SQL Server and Oracle databases.

Overview: Oracle’s High Availability Solutions

Oracle Database 12c comes with an integrated set of high availability capabilities (ref. Fig. 1) that help organizations minimize the various kinds of downtime that can affect their businesses. The next few sections provide an overview of these capabilities. For further details on each of these capabilities, please refer to [1] and [2].



Oracle Maximum Availability Architecture

Production Site

RAC
– Scalability
– Server HA

Flashback
– Human error correction

Application Continuity
– Application HA

Global Data Services
– Service Failover / Load Balancing

Active Replica

Active Data Guard
– Data Protection, DR
– Query Offload

GoldenGate
– Active-active
– Heterogeneous

Edition-based Redefinition,
Online Redefinition, Data Guard, GoldenGate
– Minimal downtime maintenance, upgrades, migrations

RMAN, Oracle Secure Backup
– Backup to tape / cloud

FIGURE 1: Integrated High Availability Features Of Oracle Database 12c

Note

Oracle Multitenant, a new option for Oracle Database 12c, delivers groundbreaking technology for database consolidation and cloud computing. The Multitenant architecture drives down IT costs by enabling a true 'manage-as-one' architecture for consolidation and virtualization of the database tier. The Multitenant architecture also makes extreme high availability a fundamental requirement when database consolidation is applied to business-critical applications. By definition, database consolidation is an exercise of 'putting all eggs in one basket.' The more successful you are at driving down cost through consolidation, the more eggs are in a single basket, and the greater is the operational and financial impact to the business should an outage occur.

New high availability (HA) capabilities in Oracle Database 12c are designed to provide the extreme level of availability required for consolidating databases onto Private Clouds. This includes support for multitenant architecture across all Oracle HA features, new levels of redundancy, transparent failover of in-flight transactions, zero-data loss disaster protection at any geographic distance. The Oracle Multitenant architecture represents the next-generation in database technology, and long-standing and time-proven Oracle HA design principles are ready from day one to provide the extreme availability required by consolidated environments.

Minimizing Unplanned Downtime

To protect from server failures, Oracle offers Real Application Clusters (RAC), which allows multiple servers to access a single Oracle database in a clustered environment. A benefit of this approach is scalability and high availability without requiring application code changes.

To protect against data failures of various kinds – e.g. those that result from storage failures, human errors, corruptions and site failures, Oracle database offers a suite of features. One of them is Automatic Storage Management (ASM), which offers integrated volume manager capabilities for Oracle. ASM provides native mirroring of database files for extra protection. To protect from human errors, Oracle database offers the Flashback suite of features (e.g. Flashback Database, Flashback Table, etc.) with which it is very easy to rewind the state of the database to a known safe point in time, and undo the effects of human errors without requiring long downtimes.

For protection of data from various media failures, Oracle database offers Recovery Manager (RMAN), which is a comprehensive backup, restore and recovery solution for the Oracle database. With RMAN, backups of the Oracle database can be taken online, without requiring expensive downtime. Furthermore, Oracle Database offers the Fast Recovery Area, which is a unified disk-based storage location for all recovery-related files and activities in an Oracle database. The automation and integration between RMAN and Fast Recovery Area provide an enhanced disk-based backup and recovery solution integrated with the Oracle Database. In addition, Oracle offers a tape and cloud backup solution as well – namely, Oracle Secure Backup (OSB), and Oracle Secure Backup Cloud Module. OSB is integrated with RMAN, providing performance optimizations and faster tape backups not available with other solutions. The cloud backup module allows administrators to use the familiar RMAN interface to backup Oracle database data changes over the cloud to Amazon S3 storage.

To protect from site or storage failures that could result from localized or regional disasters – such as fire, earthquakes, hurricanes, malicious acts, etc., Oracle offers Data Guard. Data Guard also protects against server failure for configurations where Oracle RAC has not yet been deployed. In a Data Guard configuration, multiple standby databases are connected to, and kept synchronized with, the production or primary database over a network. In the event of an unforeseen disaster at the primary data center, the processing can be easily switched to one of the standby databases, with no data loss, if desired. Data Guard standby databases can routinely be utilized for activities such as reporting, backups and quality assurance testing, without compromising data protection. Through the Active Data Guard option, this reporting could be done in a real-time manner using physical standby databases, and those databases could also be utilized for fast incremental backups. This enables active utilization of all Data Guard standby databases, and an instant ROI on the Data Guard investment.

Beyond Data Guard's disaster recovery (DR) solution, Oracle offers Oracle GoldenGate, focused on information sharing and data integration. GoldenGate allows data changes to be distributed from one or more source databases – which could be Oracle or non-Oracle, to one or more target databases, which also could be Oracle or non-Oracle. GoldenGate offers flexible capabilities such as data subsetting, data transformations, multi-master replication, active-active configurations with conflict detection, zero downtime upgrades, etc.

Oracle Database 12c introduces two new innovative features that extend the benefits of HA to the application layer. One is called Application Continuity that masks outages from end users and applications by transparently replaying the in-flight database sessions following recoverable outages within the infrastructure. The other new feature, Global Data Services (GDS) is both an availability and scalability solution that enables workload routing, connect-time and run-time load balancing and application service orchestration across replicated databases (using replication technologies such as Active Data Guard and GoldenGate).

Minimizing Planned Downtime

Planned downtime, which includes activities such as routine operations, periodic maintenance, and new deployments, can be just as disruptive to operations, especially in global enterprises that support users in multiple time zones. As with minimizing unplanned downtime, Oracle database offers a suite of capabilities that help eliminate or minimize planned downtime.

With the Online Table Redefinition capability, Oracle database supports many data maintenance operations without disrupting database operations or users updating or accessing data. For example, database tables can be redefined – changing table types, adding, dropping or renaming columns, changing storage parameters, etc. – all without interruption to end-users who are viewing or updating the underlying data. With the Rolling Upgrades capability, using Data Guard or GoldenGate, upgrades of database patchsets or major releases can be done in a rolling manner, minimizing application downtime. Patches can be installed on running Oracle instances in a fully online manner using the Online Patching capability. With the Edition-based Redefinition capability, application upgrades can be facilitated with minimal downtime by enabling the upgrades of the database components of the application while they are in use.

The Oracle Database dynamically accommodates changes to hardware configurations such as adding and removing processors from an SMP server, adding and removing nodes in a RAC cluster, dynamically growing and shrinking its shared memory allocation, adding and removing database disks online without disrupting database activities using ASM, etc. Finally, Data Guard or GoldenGate can be used for minimizing downtime during large-scale migrations such as data center moves, SAN migration, technology refresh, etc.

HA Comparison: Oracle and SQL Server

Microsoft SQL Server, with its 2012 release, introduced AlwaysOn solutions to address HA and DR requirements. Major features are AlwaysOn Failover Cluster Instance which addresses instance failover within a cluster and AlwaysOn Availability Group to failover a set of databases. Even with these new capabilities, SQL Server cannot match the depth and breadth of Oracle's high availability capabilities. This paper takes each high availability challenge and compares how Oracle Database and Microsoft SQL Server 2012 address the challenge, demonstrating how SQL Server still continues to lag Oracle significantly in this regard.

In the rest of this document, *Oracle* refers to Oracle Database 12c Enterprise Edition, and unless otherwise stated, *SQL Server* refers to Microsoft SQL Server 2012 Enterprise Edition. Descriptions of Oracle Database 12c features are available at the Oracle Database 12c documentation site [3]. All references to SQL Server 2012 are based on the Microsoft SQL Server 2012 Books Online Documentation [4].

For easy reference, the following table provides a list of the major differentiators between Oracle and SQL Server for every category of downtime. These differentiators are described in further details in the rest of the document.

TABLE 1: KEY HIGH AVAILABILITY DIFFERENTIATORS – ORACLE VS. SQL SERVER

ADDRESSING SYSTEM FAILURES	ORACLE	SQL SERVER
Active-active clustering	Yes	No
Transparent application scalability	Yes	No
Dynamic addition/removal of nodes with no effects on data distribution	Yes	No
Integrated clustering technology for all major OS and server platforms	Yes	No
Automatic workload management that enables enterprise grids	Yes	No
Recovery Advisories	Yes	No

ADDRESSING DATA FAILURES	ORACLE	SQL SERVER
Built-in database failure detection, analysis, and repair	Yes	No
Automated disk backup management	Yes	No
Built-in Incremental backup strategy	Yes	Partial
Incrementally updated backup strategy	Yes	No
Parallelize backup within a single file	Yes	No
Unused block compression during backups	Yes	No
Flexible backup compression levels	Yes	No
Automatic data file creation during recovery	Yes	No
Automatic restore failover to next available backup during recovery	Yes	No
Restore preview	Yes	No
Trial recovery	Yes	No
Clone database directly over the network, without intermediary storage	Yes	No
Integrated Mirroring	Yes	No
Fine-grained Table level recovery	Yes	No
Proactive disk health checks with automatic corruption repair	Yes	No

Cross-Platform backup & restore	Yes	No

ADDRESSING DISASTER RECOVERY – DATA PROTECTION AND AVAILABILITY	ORACLE	SQL SERVER
Standby apply progress has no impact on primary database performance or data protection	Yes	No
Silent corruptions due to hardware/software faults are detected at both primary and standby	Yes	No
Corrupt blocks are automatically repaired online, transparent to users and applications	Yes	No
Fast recovery from human error and logical corruptions	Yes	No
Integrated automatic database failover with guarantees of zero data loss and no split-brain	Yes	No
Controlled automatic failover for ASYNC to comply with user configurable data loss SLA's	Yes	No
Connect-time failover of applications in all cases – controlled by database role	Yes	No
Fast reinstatement of primary after failover without needing a full restore	Yes	No
Flexible support of multiple standbys	Yes	No
Real-time cascaded standbys	Yes	No
Database rolling upgrades across major versions and subversions	Yes	No
Support for some mixed primary/standby configurations	Yes	No
Integrated redo transport compression for efficient network utilization	Yes	No
Fast, non-disruptive recovery from network and standby database outages	Yes	No
Replicate stored procedures to standby database	Yes	No
Support database partitioning	Yes	No
Complete support of active-active database clusters	Yes	No
Long-distance, zero-data loss standby	Yes	No
Proactive readiness health checks on standbys	Yes	No

ADDRESSING DISASTER RECOVERY ROI - STANDBY DATABASE UTILIZATION	ORACLE	SQL SERVER
Full read consistency for queries on active standby	Yes	No
No datatype limitations for active standby	Yes	No
Continuous user access to an active standby database during apply of DDL changes	Yes	No
Continuous user access to an active standby database during resynchronization	Yes	No
Automatic monitoring and enforcement of query SLAs	Yes	No
Enhanced reporting and global temporary table support on active standby	Yes	No
Complete support between active standby and active-active database clusters	Yes	No
Automatic memory management on active standby	Yes	No
Dual-purpose standby database for Dev/Test	Yes	No
Integrated Load Balancing and Workload Management across replicated configurations	Yes	No

ADDRESSING HUMAN ERRORS	ORACLE	SQL SERVER
Built-in capability to mine logs and audit changes using a SQL interface	Yes	No
Built-in capability to unwind granular transactions	Yes	No
Built-in capability to ciew changes across row versions	Yes	No
Built-in ability to unwind a table to a point in time in the past	Yes	No
Built-in ability to unwind the database to a prior point in time without restoring a backup	Yes	No
Built-in capability to recover dropped objects	Yes	No
Support Recycle Bin	Yes	No
Flexible tablespace point-in-time recovery	Yes	No

ADDRESSING SYSTEM MAINTENANCE	ORACLE	SQL SERVER
Simple online addition of cluster nodes that requires no data redistribution	Yes	No
Automatic rebalance with online adding or dropping of disks	Yes	No

Online patching	Yes	No
Rolling database upgrades for full patch-sets and major releases	Yes	No
Extensive support to adjust memory online	Yes	No
Most configuration parameters may be modified online	Yes	No

ADDRESSING DATA MAINTENANCE	ORACLE	SQL SERVER
Online add, drop, exchange, move partitions	Yes	No
Online reorganization of individual tables, including relocating table to a different tablespace	Yes	No
Online reorganization of individual table partitions	Yes	No
Extensive online table redefinition capabilities, including data transformations	Yes	No
Fast online add column, with default value	Yes	No
Online rename and merge columns	Yes	No
Invisible indexes	Yes	No
Online add/modify constraint, add column, index create/rebuild do not require exclusive lock	Yes	No
DDL operations wait for user-specified time, if underlying resource is busy	Yes	No

Oracle vs. SQL Server – Addressing Unplanned Downtime

Addressing System Failures

System failures are the result of hardware failures, power failures, and operating system or server crashes. The amount of disruption these failures cause depends upon the number of affected users, and how quickly service is restored. The challenges with system failures lie in ensuring fast recovery, or better still, a higher level of fault tolerance.

As shown in the following table, Oracle provides an array of features that clearly differentiate Oracle from SQL Server in terms of how effectively it addresses system failures

TABLE 2: ADDRESSING SYSTEM FAILURES – ORACLE VS. SQL SERVER

ADDRESSING SYSTEM FAILURES	ORACLE	SQL SERVER
Active-active clustering	Yes	No
Transparent application scalability	Yes	No
Dynamic addition/removal of nodes with no effects on data distribution	Yes	No
Integrated clustering technology for all major OS and server platforms	Yes	No
Automatic workload management that enables enterprise grids	Yes	No
Recovery Advisories	Yes	No

The following sections provide further details on these differentiators.

Fast-Start Fault Recovery

The Oracle Fast-Start Fault Recovery scheme is designed to minimize downtimes related to system failures. It has two components – Fast-Start Checkpointing that optimizes roll forward recovery by continually and incrementally advancing the checkpoint position, and Fast-Start Rollback that eliminates the delays associated with the rollback phase of recovery.

Fast-Start Checkpointing – Predict Average Recovery Time

To control the time to recover from system failures, Oracle allows Mean Time To Recover (MTTR) to be directly specified via a dynamic parameter, FAST_START_MTTR_TARGET. Oracle continuously estimates the recovery time and automatically adjusts the checkpointing rate to meet the target recovery time [10].

Beyond this, Oracle takes it one step further by providing real-time feedback on the cost of the target MTTR through the `v$instance_recovery` dynamic view, as well as a GUI-based advisory through Oracle Enterprise Manager. Oracle also provides an advisory through the `v$mttr_target_advice` view that simulates the cost of a range of recovery scenarios. The simulation runs in real-time based on the current production workload. Based on the output of the advisory the administrator can choose the best tradeoff between very fast recovery time and extra I/O overhead. This takes the guesswork and risk out of configuring for fast recovery.

Fast-Start Rollback – Shorten Worst-Case Recovery Time

Oracle's crash recovery time is immune to long transactions, because Oracle allows users to access the database before instance recovery rollback operation is complete through a unique on-demand rollback technology. With Oracle, once the roll-forward processing completes, the database opens for user access. Oracle does not wait until all transactions have been rolled back. Instead, transactions are rolled back in the background while new user transactions access the data. If one of these new user transactions encounters data that was locked by a dead transaction, the user transaction instantly rolls back the change to the data made by the dead transaction and continues.

Fault Tolerance with Real World Clustering

The cornerstone of Oracle's high availability solutions that protects from system failures is Oracle Real Application Clusters (RAC). Oracle RAC is a cluster database with a shared cache architecture that overcomes the limitations of traditional shared-nothing and shared-disk approaches, to provide a highly scalable and available database solution for all business applications.

RAC supports the transparent deployment of a single database across a cluster of active servers, providing fault tolerance from hardware failures or planned outages. RAC supports mainstream business applications of all kinds –these include packaged products such as Oracle E*Business Suite, PeopleSoft, Siebel, SAP, as well as custom applications. RAC provides very high availability for these applications by removing the single point of failure with a single server. In a RAC configuration, all nodes are active and serve production workload. If a node in the cluster fails, the Oracle Database continues running on the remaining nodes. Individual nodes can also be shutdown for maintenance while application users continue to work. RAC is integrated with mid-tier clients such as Oracle JDBC, Oracle Data Provider for .NET and Oracle Call Interface (OCI) to enable automatic and coordinated connection-pool and application failover to surviving nodes, in the event of individual node failures.

A RAC configuration can be built from standardized, commodity-priced processing, storage, and network components. RAC also enables a flexible way to scale applications, using a simple scale-out model. Using sophisticated load-balancing algorithms (e.g. runtime connection pool load-balancing integrated with server-side load-balancing advisories), user sessions can be routed to the least loaded node in the cluster. On top of that, RAC supports mixed workload environments, enabling the same database to be shared by various kinds of Online Transaction Processing (OLTP) or Decision Support System (DSS) applications. Using a "Services" concept, RAC provides a simple solution to the challenges of managing different application workloads on the same database – DBAs have the power to control which processing resources are allocated to which Service during both normal operations and in response to failures. Resource allocations to Services can be made easily and dynamically enabling flexible enterprise grid environments.

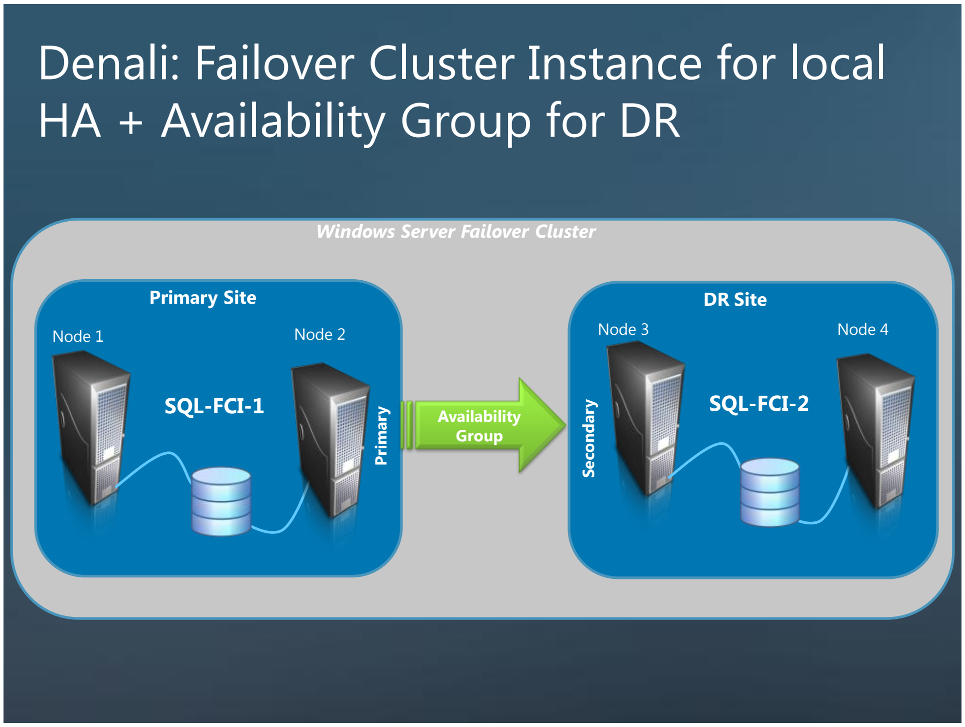
Oracle Automatic Storage Management (ASM) and Oracle Clusterware complement RAC, providing an integrated storage management and cluster software solution for enterprise grids. RAC does not have any OS or hardware restrictions, which has led to RAC being deployed at more than ten thousand customer sites around the world.

SQL Server AlwaysOn Failover Cluster Instances (FCI)

A SQL Server failover cluster instance (FCI) is a single SQL Server instance that runs in a failover cluster that consists of multiple Windows Server Failover Clustering (WSFC) nodes, thus providing HA through redundancy at the instance level. It can be used also to provide remote DR by deploying a multi-subnet FCI. On the network, an FCI appears to be an instance of SQL Server running on a single computer, but the FCI provides failover from one WSFC node to another if the current node becomes unavailable.

An availability replica can be hosted by either a standalone instance of SQL Server or an FCI instance. This allows FCI to be used for local instance-level HA and Availability Groups for database-level DR.

The diagram depicted below provides an overview of how this architecture may look like. Note that this may give the impression that this similar to Oracle RAC + Data Guard. That is not correct – simply because the secondary nodes in an FCI are all passive – they don't start their respective SQL Server instances in steady-state and are offline. In an FCI, a secondary node starts its SQL Server instance only when the resource group ownership is transferred to it during an FCI failover.



Denali: Failover Cluster Instance for local HA + Availability Group for DR

Fig. 2: SQL Server Failover Cluster Instances (FCIs) and Availability Groups

There is another serious restriction. SQL Server Failover Cluster Instances do not support AlwaysOn automatic failover for Availability Groups, so any availability replica that is hosted by an FCI can only be configured for manual failover.

FCI and WSFC

Basic FCI, which is used for local HA for the instances, is similar to legacy cold cluster (Active-Passive) architecture. Microsoft has had this basic technology for a while. For example, in SQL Server 2008, this technology was called SQL Server 2008 Failover Clustering (ref. [10]), leveraging Windows Server Failover Cluster.

In SQL Server 2012, Microsoft has grouped this basic technology under the AlwaysOn brand. How this works is quite straightforward. An FCI runs in a WSFC resource group with one or more WSFC nodes. When the FCI starts up, one of the nodes assume ownership of the WSFC resource group and brings its SQL Server instance online. At any time, only the resource group owner (and no other node in the FCI) is running its respective SQL Server services in the resource group – this is the classic Active-Passive configuration. When a failover occurs, (automatic / planned), the following sequence of events happen:

1. Unless a hardware or system failure occurs, all dirty pages in the buffer cache are written to disk.
2. All respective SQL Server services in the resource group are stopped on the active node.
3. The resource group ownership is transferred to another node in the FCI.
4. The new resource group owner starts its SQL Server services.
5. Client application connection requests are automatically directed to the new active node using the same virtual network name (VNN).

SQL Server has no solution equivalent to RAC.. To address scale-out architecture, apart from FCI, SQL Server also has some primitive methods such as Distributed Partitioned Views where the data is horizontally partitioned across servers, Scalable Shared Databases which are read-only instances and so on. These models lead to complexities in the areas of data partitioning (to avoid "hot nodes"), adding/removing nodes, dealing with node failures, etc.

It may be noted that to protect from server failures, Microsoft mandates using SQL Server with Microsoft Cluster Service (MSCS), in a Failover Clustering model [6], [7]. However, in this model, a particular SQL Server instance runs in only one node, while the other "backup" node remains in a passive state, waiting for the failover to occur. The SQL Server AlwaysOn FCI single-site deployment provides some capabilities similar to *Oracle RAC One* in a shared storage mode, in which only one node is active. It can be deployed in a multi-site cluster model too. However, in contrast to Oracle RAC's active-active model where all nodes serve production workload, this is a traditional cold-cluster active-passive deployment, leading to under-utilization of systems resources and higher recovery times.

Failover Cluster Instance – Nothing Like RAC

The SQL Server and Windows Server Clustering technology is still based on legacy Active-Passive database clustering model, leading to wastage of systems resources. This is primitive compared to the active-active database clustering model (Real Application Clusters) that Oracle had pioneered more than 10 years back.

For further details on the technical merits of Oracle RAC offering an integrated high availability and scalability solution compared to similar offerings from Microsoft, refer to [8].

Addressing Data Failures

It is vital to design a solution to protect against, and recover from, data and media failure. A system or network fault may prevent users from accessing data, but media failures without proper backups can lead to lost data that cannot be recovered.

As shown in the following table, Oracle offers a wide-ranging set of capabilities to address data failures, which differentiates itself from SQL Server.

TABLE 3: ADDRESSING DATA FAILURES – ORACLE VS. SQL SERVER

ADDRESSING DATA FAILURES	ORACLE	SQL SERVER
Built-in database failure detection, analysis, and repair	Yes	No
Automated disk backup management	Yes	No
Built-in Incremental backup strategy	Yes	Partial
Incrementally updated backup strategy	Yes	No
Parallelize backup within a single file	Yes	No
Unused block compression during backups	Yes	No
Flexible backup compression levels	Yes	No
Automatic data file creation during recovery	Yes	No
Automatic restore failover to next available backup during recovery	Yes	No
Restore preview	Yes	No
Trial recovery	Yes	No
Clone database directly over the network, without intermediary storage	Yes	No
Integrated Mirroring	Yes	No
Fine-grained Table level recovery	Yes	No
Proactive disk health checks with automatic corruption repair	Yes	No
Cross-Platform backup & restore	Yes	No

Much of the brains behind Oracle's data failure protection lie with Oracle Recovery Manager (RMAN), which is a database-integrated technology to facilitate efficient backup and recovery of Oracle databases. RMAN optimizes performance and space consumption during backup with file multiplexing and compression. It also ensures that backups are corruption-free by validating production data block integrity during backup operations and validating integrity of backups when they are restored. RMAN leverages the Fast Recovery Area and integrates with Oracle Secure Backup [9] and third party tape backup management products for a centralized disk-to-disk-to-tape backup strategy.

The following sections provide further details on the Oracle differentiators mentioned in the above table.

Built-in Database Failure Detection, Analysis, and Repair

When faced with data failures, a DBA first invests time to diagnose the issues and plan an appropriate recovery strategy. Depending on the nature of the failure, this investigation and planning time can often comprise a large percentage of the total recovery time. Available with Oracle Database 12c, the Data Recovery Advisor (DRA) dramatically reduces this time by automatically detecting failures in real-time (e.g. block corruptions, missing files), reporting failure analysis results, and generating a feasible recovery strategy (e.g. RMAN recovery script) that can be run as-is or customized for running at a later time. In addition, regularly scheduled Data Integrity Checks allow proactive monitoring of database integrity, thereby catching and repairing data issues before users even come across them.

In large environments where DBAs manage hundreds of databases and thousands of data files, the DRA can dramatically simplify recovery diagnosis and management tasks. By self-diagnosing Oracle failures, the DRA also reduces the chances that a user could develop an improper recovery strategy or commit errors, while under the pressure of recovering a critical production system.

SQL Server has no such intelligent, database-aware diagnosis and recovery tool and continues to rely on manual restore, recovery, and data verification procedures.

Automated Disk Backup Management

Disk-based backups provide higher performance and reliability over traditional direct-to-tape backup. With lower cost disk, such as SATA drives, backups on disk now become the primary source for recovery while tape backups are used primarily for archival purposes. The Fast Recovery Area (FRA) consolidates all RMAN backups and recovery related files on disk into a single storage location on a filesystem or Automatic Storage Management (ASM) disk group. Oracle monitors and manages the space usage, automatically removing obsolete backups to make room for the latest database backups.

SQL Server does not offer automated disk backup management. The only backup obsolescence method is to specify a time duration beyond which all files in the backup location expire and consequently can be deleted. A DBA has to manually separate the files that are needed from those that are not, in case the system is running out of space and unnecessary files need to be removed. If the SQL Server instance consists of multiple databases (similar to Oracle tablespaces), the DBA's management burdens are further compounded, not to mention increasing the possibility of user error, as all needed files and backups must now be tracked and maintained across multiple databases to ensure successful recovery.

Incremental Backup Strategy

Using incremental backup strategy, users can backup only the changes happened after the previous full or incremental backup. Oracle allows both differential (changes after the previous full or incremental backup) and cumulative (changes after the last full backup) incremental backups. Moreover, using Block Change Tracking mechanism, Oracle keeps track of the changes. This tracking mechanism dramatically reduces the backup window – since the backup doesn't have to scan the entire database. SQL Server only supports cumulative incremental backups. Note that Microsoft mentions this cumulative backup strategy as *differential backup*.

Integrated Incrementally Updated Backup Strategy

With integrated fast incrementally updated backups, RMAN rolls forward an image copy by applying incremental backups. The image copy is updated with block changes up through the SCN at which the latest incremental backup was taken. Incrementally updated backups eliminate the need and overhead of performing a full database backup every day. SQL Server does not have this capability.

Parallelize Backup within a Single File

RMAN can back up or restore a single file in parallel by dividing the work among multiple channels. Each channel backs up one file section, which is a contiguous range of blocks. This speeds up overall backup and restore performance, and particularly for BIGFILE tablespaces, in which a data file can be sized upwards of several terabytes. SQL Server does not offer a comparable capability.

Unused Block Compression

RMAN can reduce backup sizes considerably by eliminating blocks that are not currently being used, during full backups. For example, if a 1 TB table is dropped and purged, the next full backup would not backup those 1 TB worth of blocks. SQL Server does not offer this block elimination capability.

Flexible Backup Compression Levels

RMAN has offered native backup compression since Oracle Database 10g, which can achieve 40%+ reduction in backup sizes. Starting Oracle Database 11g, RMAN offers more backup compression levels to flexibly suit a particular environment's compression ratio and backup performance needs. Users can choose from the new HIGH, MEDIUM, and LOW levels, based on the desired degree of compression. SQL Server only offers one backup compression setting by default.

Automatic Data File Creation during Recovery

Automatic datafile creation enables RMAN to recreate datafiles automatically during recovery, provided that all archived logs dating back to the creation of the datafile are available. This eliminates the need to take a backup after adding a new tablespace or data file. SQL Server does not offer a similar feature.

Automatic Restore Failover during Recovery

During a restore, when RMAN finds corruption in a backup, or finds that a backup cannot be accessed, RMAN tries to restore the file from all known backups before returning an error. This is done automatically whenever RMAN restores file(s) with the RESTORE or RECOVER command, eliminating the need to search for valid backups and re-trying the operation when a restore failure occurs. SQL Server lacks this capability.

Restore Preview

Before a database restore operation, a DBA can request to view the list of backup files needed to complete the operation. The RMAN restore preview capability ensures that all required backups are available or to identify situations in which the DBA may want to direct RMAN to use or avoid specific backups. SQL Server does not offer this capability.

Trial Recovery

A test recovery can be very useful to first ensure that all required archived logs are present, corruption-free, and can be successfully applied to the restored data files, without having to perform actual media recovery. Oracle trial recovery provides just that and does not modify the restored data files. SQL Server does not provide this capability.

Database Cloning over the Network

A common DBA task is to clone production databases for test, QA, reporting, and disaster recovery purposes. A backup can be used to restore and create the clone database, but this requires the backup to be copied or made accessible to the clone server. In Oracle Database 12c, RMAN offers a method to clone an online production database over the network directly to a clone server, without requiring a backup. SQL Server does not provide this cloning capability.

Integrated Fine-grained Table Recovery

In Oracle Database 12c, RMAN can recover individual database tables from backup, via a simple `RECOVER TABLE` command. This recovers one or more tables (the most recent or an older version) from an RMAN backup. Tables can be recovered in-place or to a different tablespace. Optionally, RMAN can create a Data Pump dump file of the table(s). This efficient functionality replaces error-prone manual processes and reduces recovery time. It extends the range of recovery in areas where Flashback is not applicable, for example when a dropped table has been purged out of the Recycle Bin, or when the desired point to recover is outside the window given by the `UNDO_RETENTION` parameter. SQL Server does not have recoverability feature to this level of granularity.

Integrated Backup Encryption

Protecting backups of highly sensitive and confidential information is vital to the stability of many companies. Backups should only be able to be opened and read by their creators. RMAN provides the ability to encrypt backups as they are created, in a database-integrated manner, with 128, 256, or 512-bit versions of the Advanced Encryption Standard (AES).

Cross-platform Backup and Restore

New RMAN Cross-platform functionality in Oracle Database 12c enables backup and restore across different platforms, for the most efficient tablespace and database migration which can improve application availability. In earlier releases, moving a database across platforms required either import/export or cross-platform transportable tablespaces procedures, thereby, affecting application availability. On the source platform, BACKUP creates backup sets of user tablespaces, including Data Pump metadata dump file, in read-only mode. RESTORE on the destination platform automatically performs data file Endian conversion and plugs-in tablespaces. To minimize read-only impact, Oracle recommends taking incremental backups, that are then converted and applied to restored data files. Only the final incremental must be taken while tablespaces are in read-only mode, with separate Data Pump metadata export and import.

SQL Server is only supported in Windows environment.

Integrated Fine-grained Table Recovery

In Oracle Database 12c, RMAN can recover individual database tables from backup, via a simple `RECOVER TABLE` command. This recovers one or more tables (the most recent or an older version) from an RMAN backup. Tables can be recovered in-place or to a different tablespace. Optionally, RMAN can create a Data Pump dump file of the table(s). This efficient functionality replaces error-prone manual processes and reduces recovery time. It extends the range of recovery in areas where Flashback is not applicable, for example when a dropped table has been purged out of the Recycle Bin, or when the desired point to recover is outside the window given by the `UNDO_RETENTION` parameter.

SQL Server does not have this fine-grained recovery capability.

Additional Responses to SQL Server 2012 Backup & Restore Collateral

In the SQL Server 2012 High Availability datasheet [10], Microsoft highlights several features with respect to its backup & restore capabilities. The following sections discuss each of those features and demonstrate how Oracle has had similar, if not – more comprehensive capabilities, for several releases.

Backup Compression

The new backup feature introduced in SQL Server 2008 Enterprise Edition is support for native compression during backups [11]. This can be enabled for all backups of a particular database or on a per-backup command level via the WITH COMPRESSION clause. Any edition of SQL Server 2008 and above can restore a compressed backup. Both compression ratio and performance can be affected by data type, prevalence of redundant values across multiple rows, and whether encryption or compression has already been applied to the data.

RMAN has offered backup compression since Oracle Database 10g in both Standard and Enterprise Editions. With Oracle Database 12c, RMAN offers expanded backup compression levels to flexibly suit a customer's compression ratio and backup performance needs. Users can choose from the new HIGH, MEDIUM, and LOW levels, based on the desired degree of compression [12].

Microsoft recommends running compressed backups in a special CPU-restricted session (setup using Resource Governor) to minimize impact to concurrent operations. Oracle offers a similar ability to run compressed backups in a CPU-restricted session using Resource Manager [13].

Restore Read-only Database Filegroups without Transaction Logs

SQL Server has the ability to place filegroups into read-only mode, if its constituent tables will not be modified. Once a filegroup is made read-only and backed up, further filegroup and transaction log backups are not required, unless the filegroup is made read-mostly or read-write in the future. Thus, if a read-only filegroup needs to be restored, no transactions logs are needed for recovery. Oracle provides a similar concept in read-only tablespaces. Once a read-only tablespace is backed up, no further backups are needed unless the tablespace is made read-write or the backup expires from tape. When a read-only tablespace is restored and recovered, no archived redo logs are applied, and the tablespace can be immediately placed online afterwards.

Restore Individual Data Pages

SQL Server can restore individual data pages (similar to Oracle blocks), if they are damaged, and recover with transaction logs, while the database remains online. This is similar to RMAN block media recovery (available since Oracle9*i*), in which blocks identified as corrupt are restored from a good backup and recovered with archived redo logs, while the database remains online.

Furthermore, in Oracle Database 11g Release 2, when an Active Data Guard standby database has been configured, a corruption that occurs on the primary database will be automatically detected and transparently repaired in-line with a good block from the standby database. A user or application query accessing the corrupt block will only experience a short pause while the block is repaired and then the query results are returned.

Piecemeal Restore

In event of full or partial database recovery, this restore method allows the primary (i.e. system) and most critical secondary (i.e. user) filegroups to be restored and recovered first, so that their data can be made available, followed by subsequent restores of less critical filegroups, until all needed filegroups are available. RMAN has a similar method in which the database can be mounted (not open) and all user datafiles are marked offline. Then, system, undo, and any critical user tablespaces can be restored and recovered first. Those user tablespaces are then placed online for access. The rest of the needed tablespaces can then be subsequently restored, recovered, and placed online in a phased approach.

Online Restore

SQL Server allows the database to remain online and available during database file, page, or piecemeal restore operations. RMAN has offered online restore since Oracle8, supporting online restore at the granularity of the datafile and tablespace, and with Oracle9*i*, restore and recovery of individual blocks with block media recovery. In addition, RMAN supports online 'instant' restore by switching data files to immediately use existing image copy backups, instead of traditional restore process, which incurs time to physically copy the data files back to their original location.

Partial Database Availability

In the event that SQL Server non-primary data files are damaged, the rest of the database continues to remain open and available. Oracle fully supports offlining of user data files and this does not affect access to the rest of the database.

Mirrored Backup Sets

Mirrored backup sets allow a copy of a SQL Server backup to be made to different backup devices, to increase protection in the event of backup media failure. RMAN also supports creating a copy of a backup to different disk locations. Furthermore, as discussed previously, if RMAN encounters a missing or corrupt backup, it automatically fails over to another copy of the backup, if possible, and if not, will restore older backups until a good backup is found.

Data Page and Backup Checksum

SQL Server allows damaged database pages to be quickly detected in real-time using page checksums. A checksum is written upon a page write and is verified when a page is read. Damaged pages could result from incomplete page write or stem from storage-related issue. Online page-level restore can be used to replace the faulty page. Oracle has a comparable checksum capability where a checksum is written into the block header when a block is written and it is validated upon block read. Furthermore, Oracle can check the logical contents of the block itself, in cases where checksum validation may pass, but logical inconsistencies remain (e.g. header information is inconsistent with data, row and column information is invalid). Similarly, RMAN block media recovery can be used to perform block-level recovery, while database remains online.

SQL Server can validate page checksums during backup and write separate checksums into transaction log and backup files. Log and backup file checksums are validated on restore and recovery operations. Similarly, Oracle validates block checksums (and optionally, logical block contents) during backup and retains block checksums in the backup files. Log and backup file checksums are validated on restore and recovery.

Furthermore, RMAN can validate block checksums and logical block contents in-memory, without physically writing out the backup files. This can be useful to periodically ensure that the database is corruption-free. RMAN also supports a test restore capability in which backups can be validated, without having to physically write the data files back to their original location. Finally, as discussed previously, RMAN supports trial recovery, which allows archived logs to be verified and applied in-memory first, without physically altering the restored data files. SQL Server does not offer database validation, nor test restore and recovery procedures.

ASM – Integrated Data Mirroring

Oracle Automatic Storage Management (ASM), which is an integrated volume management and file system [14] available with the Oracle Database, provides a native mirroring mechanism based on the concept of *disk failure groups*, which can be used to protect against storage failures. An ASM failure group is a set of disks sharing a common resource (disk controller or an entire disk array) whose failure can be tolerated. With ASM mirroring, when database extents are allocated, a primary copy and a secondary copy are created, with the disk for the secondary copy chosen to be in a different failure group than the primary copy. This ensures that the data is available and transparently protected against the failure of any component in the storage subsystem. Not only that, if there are read errors associated with reading a corrupted block, ASM will transparently read a good copy of the extent, and copy it to the disk that had the read error.

SQL Server does not provide any such integrated mirroring mechanism for additional data protection. It relies on underlying storage technologies or volume manager such as Storage Spaces.

Addressing Disaster Recovery

Oracle Data Guard

Oracle Data Guard provides the management, monitoring, and automation software infrastructure to create and maintain one or more standby databases to protect Oracle data from failures, disasters, errors, and data corruptions. In the event of a planned or unplanned outage at the primary site, Data Guard ensures that a standby database can be easily switched to the production database role without data loss, client connections are automatically redirected, and the new production database commences with serving enterprise data needs. There are two types of Data Guard standby databases. A *physical standby* uses Redo Apply to maintain a block for block, exact replica of the primary database. A *logical standby* uses SQL Apply and contains the same logical information as the primary database, although the physical organization and structure of the data can be different. The advanced capabilities of Data Guard 11g - Oracle Active Data Guard and Data Guard 12c Snapshot Standby, and in Oracle Database 12c – Data Guard Far Sync, deliver the highest levels of availability, data protection, operational transparency, and return on investment (ROI) in standby software, servers, and storage.

SQL Server AlwaysOn Availability Groups

AlwaysOn Availability Group (AG) is based on a discrete set of user databases, known as availability databases, which fail over together. An availability group supports a set of primary databases (a single primary replica) and one to four sets of corresponding secondary databases (secondary replicas). An availability group fails over at the level of an availability replica, which means failovers in this context don't occur because of database-specific issues such as a loss of a data file, deletion of a database, or corruption of a transaction log. . This solution is an alternate to the database mirroring technology. Synchronous and Asynchronous replication modes are supported for DR purposes. Up to 4 secondary replica databases can be configured. The primary replica makes the primary databases available for read-write connections from clients and, also, sends transaction log records for each primary database to every secondary replica. Every secondary replica writes the transaction log records to disk (hardens the log) and applies the log records to its own set of secondary databases and serves as a potential failover target for the AG.

One or more secondary replicas can be configured to support read-only access to secondary databases, and any secondary replica can be configured to permit backups on secondary databases.

AlwaysOn AG rely on Windows Failover Clustering (WSFC) to monitor and manage the current roles of the availability replicas that belong to a given availability group and to determine how a failover event affects the availability replicas. For an instance of SQL Server to be eligible for AlwaysOn AG, the instance must reside on a WSFC node, and the WSFC cluster and node must be online. Each replica is hosted by an instance of SQL Server on a different node of the WSFC cluster.

The following diagram shows the relationship between Availability Groups and WSFC.

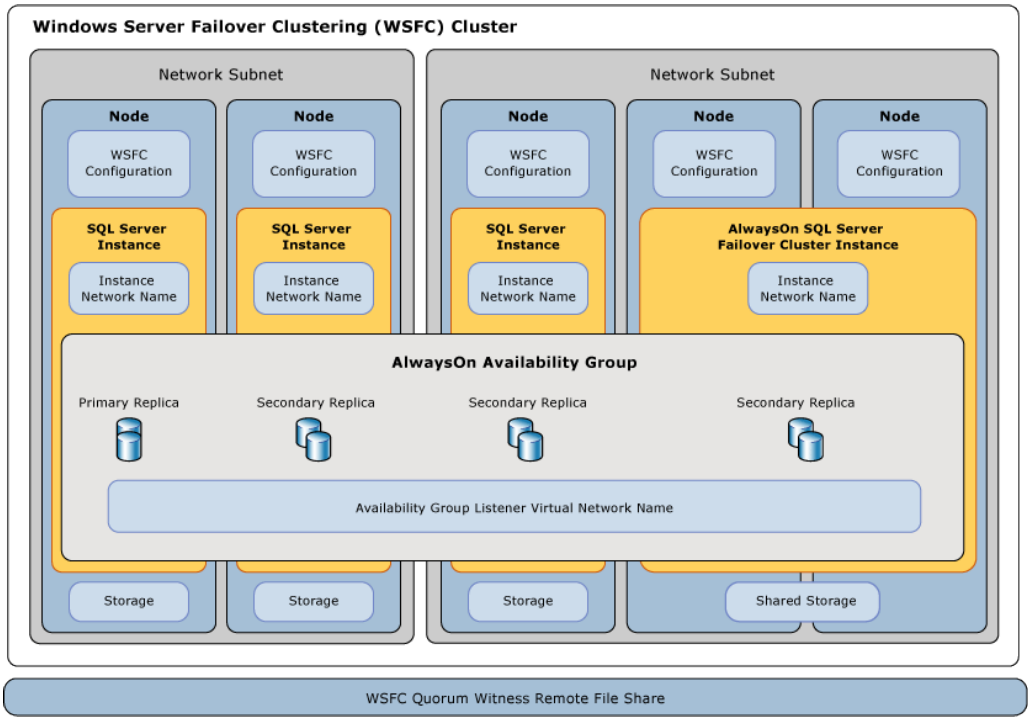


Figure 3: SQL Server Availability Group Overview

Failover: AlwaysOn Availability Group

SQL Server doesn’t use a separate Switchover term as is used in Data Guard. For SQL Server, all role transitions are called Failovers. In fact, SQL Server doesn’t support something like a Switchover for asynchronous connections.

During a SQL Server failover, the target secondary replica transitions to the primary role, becoming the new primary replica. The new primary replica brings its databases online as the primary databases, and client applications can connect to them. When the former primary replica is available, it transitions to the secondary role, becoming a secondary replica. The former primary databases become secondary databases and data synchronization resumes.

SQL Server 2012 supports three types of failovers:

1. Automatic
2. Manual
3. Forced (with possible data loss)

The failover supported by a given secondary replica depends on its availability mode, and, for synchronous-commit mode, on the *failover mode* property on the primary replica and target secondary replica.

Synchronous-commit mode supports two forms of failover – planned manual failover without data loss (e.g. used for administrative reasons), and automatic failover without data loss, and in both such cases, the secondary replica must already be in a SYNCHRONIZED state. For automatic failover, the target secondary replica must have Windows Server Failover Clustering (WSFC) quorum and the WSFC witness must be connected to the cluster node where the secondary replica is located.

Under asynchronous-commit mode, the only form of failover is forced manual failover, with possible data loss. This implies lack of a capability similar to Data Guard switchover.

SQL Server Multi-Subnet Clustering

SQL Server multi-subnet failover cluster, while supported in SQL Server 2008 R2, gets slightly better in SQL Server 2012 through better integration with features from Windows Server 2008 R2 (ref. [12]). Conceptually, this is a Stretch Cluster, with the important distinction compared to Oracle RAC Stretch Cluster being only one node is active at any given time for a given database.

In this SQL Server configuration, each failover cluster node is connected to a different subnet or different set of subnets. These subnets can be in the same location or in geographically dispersed sites. There is no shared storage that all the nodes can access, so data should be replicated between the data storage on the multiple subnets. SQL Server relies on external products such as storage-based mirroring, to provide this data replication.

The following diagram visualizes how this works. Note that in steady state, only Node1 is active, all other nodes are passive. Within Site A, Node1 and Node2 provide local HA in a cold-cluster configuration.

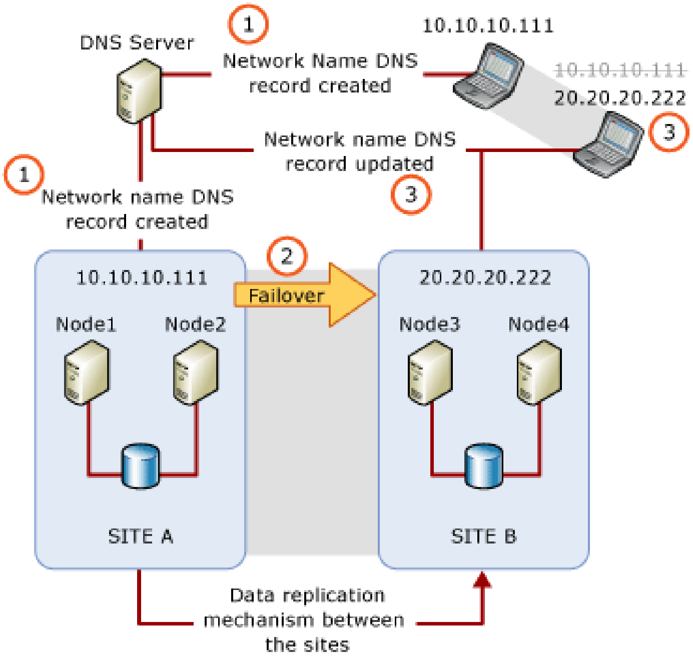

DNS Server
1
Network Name DNS record created
10.10.10.111
~~10.10.10.111~~
20.20.20.222
3
Network name DNS record updated
3
1
Network name DNS record created
2
Failover
10.10.10.111
Node1
Node2
SITE A
20.20.20.222
Node3
Node4
SITE B
Data replication mechanism between the sites

Fig. 4: Two node, Two subnet SQL Server Failover Cluster

With SQL Server 2008 implementation doesn't require a Virtual Local Area Network (VLAN) configuration as required by the previous version.

Configuring multiple subnets is simpler than VLANs to set up and manage. However, when using multiple subnets, the amount of downtime that clients experience because of the failure is dependent not just on how quickly failover occurs, but also on how quickly DNS replication occurs (between the DNS servers) and how quickly the clients query for updated DNS information (after the cached DNS entries expire on the client). Because of this, specific configuration settings such as node heartbeat settings and HostRecordTTL, as well as *intersite replication frequency* settings for Active Directory Sites and Services, are very critical for SQL Server multi-subnet clustering environments.

Note that the default Time to Live for Windows 2008 R2 Server and even Windows 2012 Server is 20 minutes, which is the downtime that SQL Server clients will face on a site failover if this setting is not adjusted. Microsoft recommends adjusting this to 1 minute. However, worse, the default DNS update replication frequency in Windows environment is 180 minutes, with 15 minutes being the minimum value (ref. [13]). This implies that even using these lower values, SQL Server clients will take minimum 16 minutes to reconnect after a failover, which will not satisfy the HA / DR SLAs of today's mission critical applications (ref. [14]).

There is a limitation with AlwaysOn Availability Group failover whereby, if there is no router exists between the failover cluster and the application server, there will be a long delay for the Cluster service to fail over network-related resources. [KB Article : 2582281]

In contrast, for multi-site HA / DR deployment of Oracle databases involving RAC and/or Data Guard, Oracle completely avoids these network complexities through database-integrated mechanisms such as Fast Application Notification (FAN) and Fast Connection Failover (FCF). For Oracle MAA configurations, it is not uncommon to find customers doing inter-site failovers in the order of a few seconds (ref. [15]).

Microsoft Database Mirroring

Microsoft Database Mirroring has similar goals as Data Guard for increasing database availability and providing disaster protection, though there are many differences in its capabilities when compared to Data Guard.

Database Mirroring is a relatively new implementation, first released in April of 2006. The immaturity of Database Mirroring causes Microsoft to continue recommending traditional Log Shipping as an alternative or complementary solution to address gaps in functionality. Both Microsoft technologies are described in the following sections.

Note: Microsoft has already warned that Database Mirroring feature will be removed from the future version of SQL Server. AlwaysOn AG which was introduced with SQL Server 2012 will be the preferred mode for replication. [http://technet.microsoft.com/en-us/library/ms189852.aspx]

Microsoft Log Shipping

Microsoft log shipping allows users to send transaction log backups from a primary database to one or more secondary databases to maintain remote replica(s) that is a “warm standby” of the production database. Microsoft log shipping is similar to the "Managed Standby" feature in Oracle8*i*, an early predecessor to Oracle Data Guard 12c. Log shipping has several disadvantages:

- Data loss potential is significant since all current log data and any log backups that have not been transmitted to the standby database will be lost should the primary fail.
- Log shipping requires a third server instance, known as the monitor server, if the customer wishes to record the history and status of backup and restore operations and, raise alerts if these operations fail to occur as scheduled.
- The operation and management of log shipping is not integrated with Database Mirroring, introducing further complexity if both mechanisms are to be used in a complementary fashion.

Data Guard & AlwaysOn AG Comparison

Data Guard is a comprehensive solution able to function in challenging operating environments while delivering superior functionality. Compared to the complex AlwaysOn AG setup, a simple multi-standby Data Guard configuration provides greater management simplicity, better data protection, and higher availability; all while productively utilizing all standby databases to achieve maximum ROI.

The following table summarizes the comparative strengths of Data Guard over SQL Server AlwaysOn AG solution.

TABLE 4: ADDRESSING DISASTER RECOVERY – ORACLE VS. SQL SERVER

ADDRESSING DISASTER RECOVERY – DATA PROTECTION AND AVAILABILITY	ORACLE	SQL SERVER
Standby apply progress has no impact on primary database performance or data protection	Yes	No
Silent corruptions due to hardware/software faults are detected at both primary and standby	Yes	No
Corrupt blocks are automatically repaired online, transparent to users and applications	Yes	No
Fast recovery from human error and logical corruptions	Yes	No
Integrated automatic database failover with guarantees of zero data loss and no split-brain	Yes	No
Controlled automatic failover for ASYNC to comply with user configurable data loss SLA's	Yes	No
Connect-time failover of applications in all cases – controlled by database role	Yes	No
Fast reinstatement of primary after failover without needing a full restore	Yes	No
Flexible support of multiple standbys	Yes	No
Real-time cascaded standbys	Yes	No
Database rolling upgrades across major versions and subversions	Yes	No
Support for some mixed primary/standby configurations	Yes	No
Integrated redo transport compression for efficient network utilization	Yes	No
Fast, non-disruptive recovery from network and standby database outages	Yes	No
Replicate stored procedures to standby database	Yes	No
Support database partitioning	Yes	No
Complete support of active-active database clusters	Yes	No

Long-distance, zero-data loss standby	Yes	No
Proactive readiness health checks on standbys	Yes	No

ADDRESSING DISASTER RECOVERY ROI - STANDBY DATABASE UTILIZATION	ORACLE	SQL SERVER
Full read consistency for queries on active standby	Yes	No
No datatype limitations for active standby	Yes	No
Continuous user access to an active standby database during apply of DDL changes	Yes	No
Continuous user access to an active standby database during resynchronization	Yes	No
Automatic monitoring and enforcement of query SLAs	Yes	No
Enhanced reporting and global temporary table support on active standby	Yes	No
Complete support between active standby and active-active database clusters	Yes	No
Automatic memory management on active standby	Yes	No
Dual-purpose standby database for Dev/Test	Yes	No
Integrated Load Balancing and Workload Management across replicated configurations	Yes	No

Disaster Recovery – Data Protection and Availability

Data Guard architecture and capabilities provide superior data protection and data availability compared to SQL Server 2012 AlwaysOn as described below:

Detecting silent corruptions caused by lost writes

Data Guard provides industry-unique protection against lost writes. A lost write occurs when an I/O subsystem acknowledges to Oracle that a write is complete, while in fact the write did not occur in the persistent storage. On a subsequent block read, the I/O subsystem returns the stale version of the data block, which can be used to update other blocks of the database, thereby corrupting it. Oracle provides a `DB_LOST_WRITE_PROTECT` initialization parameter which when set, will record buffer cache block reads in the redo log and use this information to detect lost writes. When a Data Guard standby database applies this redo during managed recovery, it reads the corresponding blocks and compares the SCNs with the SCNs in the redo log to determine if there has been a lost write. The procedure recommended for repairing lost writes on a primary database is to failover to the physical standby and recreate the primary. If the lost write has occurred on the standby, simply recreate the standby database or the affected files. SQL Server does not have the ability to detect lost writes and avoid data loss and production downtime that results from such an event.

Built-in mechanisms to undo corruptions related to human errors

Human error is one of the leading causes of downtime. Such errors lead to logical corruptions that can be widespread and result in significant downtime for point-in-time recovery operations. Data Guard can quickly recover from such corruptions using delayed apply – which delays the processing of redo on the standby by a configurable period of time. This provides administrators the opportunity to failover to the standby before the error is applied to the standby database, or to export still valid data from the standby and use it to surgically repair the primary database. A typical example of such an error is an incorrect batch job run on the primary database.

A compromise that accompanies using a delay is that the standby database must first apply the backlog of log data before it can assume the primary database role – increasing downtime should a failover be required. Oracle Flashback Database offers Data Guard users a second method to achieve the same level of protection by enabling the primary and standby databases to be quickly recovered to a previous point in time. Flashback Database eliminates any delay at failover time because the standby database is always up to date. Flashback Database is the recommended approach when also using an Active Data Guard standby for up-to-date queries and reporting.

SQL Server falls short of Oracle Flashback and Oracle Data Guard to address this problem.

Integrated automatic database failover with zero data loss

Data Guard Fast-Start Failover automatically detects primary database outages and executes failover to a previously chosen, synchronized standby database – no manual intervention or external integration with clusterware software is required. Once the original primary has been repaired, mounted, and is able to establish a connection with the new primary, Data Guard automatically converts the failed primary to a standby database and resynchronizes it without requiring a time-consuming restore from backup (both SYNC and ASYNC configurations).

Fast-Start Failover also guarantees that automatic failover can never result in “split brain” (a condition when 2 or more databases in a primary/standby configuration act as primaries simultaneously). It also supports advanced capabilities such as doing such automatic failovers even in asynchronous transport conditions (as long as RPO SLAs are not violated), or upon designated health check violations.

SQL Server AlwaysOn AG relies on Windows Failover Cluster to perform failovers under certain fault conditions.

Connect-time failover of applications in all cases

Data Guard combined with Oracle Transparent Application Failover (TAF) enables client-side connect time failover using an alternate connect descriptor. Oracle Data Guard 11g Release 2 implements role-specific database services – ensuring that services are automatically started and are appropriate for the then-current role of the database (primary, standby, or snapshot standby). Automatic client failover in a Data Guard configuration is not dependent upon the client having already connected to a primary database in order to determine the failover partner. This is important in situations where the primary database is unavailable to new client connections or in configurations having multiple standby databases, any one of which is able to assume the role of primary database at failover time.

SQL Server relies on Windows Failover Cluster to perform failovers during any timeout conditions.

Easy reinstatement of primary to a new standby database after a failover

Data Guard supports fast reinstatement of a failed primary as a new standby database without requiring a full restore from backup – for both SYNC and ASYNC failovers. This quickly returns the configuration to a protected state, practically eliminating any extra effort on the part of DBAs.

SQL Server requires a full restore from backup any time that the primary and standby databases are not completely synchronized at failover time.

This can be a daunting task given the common occurrence of multi-terabyte databases and WAN topologies where high latency or bandwidth limitations make a remote restore even more time consuming and expensive. Several undesirable outcomes result from this limitation:

- There is a tendency to failover to the standby database only as a last resort after significant downtime has elapsed.
- There is a high network cost – disrupting other applications that share the same network bandwidth, while a backup of the new primary is transmitted to the original primary site.
- Time, effort, frustration, and potential for human error during the restore process, often result in DBA’s expending even more time, effort, and frustration.
- There is an extended period of time when the new primary database is in an unprotected state and thus is vulnerable to data loss and downtime should there be a second failure event.

Real-time cascaded standby(s)

To reduce the load on the primary system, Data Guard supports cascading standbys. A cascading standby database is a standby database that receives it redo logs from another standby database, not from the original primary system. The standby database would send redo to the cascaded standby upon a log switch. Oracle Database 12c has taken cascaded standby databases to the next level providing real-time cascading capabilities in that redo logs are immediately forwarded from the standby to the cascaded standby instead of waiting for a log switch to occur.

SQL Server does not have similar feature.

Support for mixed primary/standby configurations to reduce planned downtime

Data Guard has the flexibility to support a number of configurations where primary and standby systems may have different CPU architectures, operating systems (e.g. Windows and Linux), operating system binaries (32-bit/64-bit), and Oracle database binaries (32-bit/64-bit). This offers a method of reducing planned downtime and risk when executing certain platform migrations and technology refresh.

Data Guard can also be used to migrate to Automatic Storage Management (ASM), from single instance Oracle Databases to Oracle RAC, or to Oracle Exadata storage.

Finally, a Data Guard standby database can be configured with fewer resources (e.g. CPU, memory, I/O) than its primary database. This provides customers flexible deployment options. For example, one deployment model is to consolidate the hosting of multiple standby databases on a single server or Oracle RAC. This can reduce the investment required in standby systems if such a model meets business objectives.

SQL Server does not have such flexibility. While SQL Server is only supported with Microsoft Windows environment, other planned downtime tasks such as storage migration etc. causes downtime.

Integrated network transport compression

Data Guard 11g Release 2 with the Oracle Advanced Compression Option enables automatic compression of all redo transmission between primary and standby databases. Transport compression enables more efficient utilization of available network bandwidth. This enables customers with limited bandwidth to achieve their recovery point objectives even if their redo volume exceeds available bandwidth. It also results in faster resynchronization of primary and standby databases following any outage that interrupts network transmission.

SQL Server does not provide an integrated capability for network compression

Fast, non-disruptive recovery from replication and standby database outages

In addition to transport compression, Data Guard is well architected for high-performance log gap resolution to quickly recover from any outage that impacts replication, such as network or standby database outages. Multiple parallel Data Guard background processes automatically transmit large volumes of archive log data that may be required to resynchronize a standby database following an extended outage. While this occurs in the background, Data Guard also transmits current log file data thereby preventing transmission from falling any further behind. High volume, parallel shipment is possible due to the high degree of isolation between primary and standby systems in a Data Guard configuration. If there is a large backlog of data to be transmitted, a Data Guard standby database is able to receive data faster than it can be processed without any negative impact on network transport, primary database performance or availability.

The faster the data can get to the standby, the faster the data is protected and the sooner the database returns to a protected state.

SQL Server does not have an equivalent capability.

Complete support for active-active database clusters

Data Guard is completely integrated with Oracle RAC. Any of the primary or the standby databases can be multi-node Oracle RAC instance. All Data Guard protection modes, transport modes, and apply modes are supported. Automated transmission of redo data and recovery are available for all configurations. Any physical standby database, whether Oracle RAC or single node, can also be an Active Data Guard Standby database and support read-only queries while it applies updates received from its primary database. Data Guard and Oracle RAC support all platforms that are supported by the Oracle Database.

SQL Server does not support active-active database.

Long-distance, Zero-data Loss Standby

Oracle Database 12c introduces Data Guard Far Sync functionality that enables extending the zero-data loss standby protection to any distance – no longer limited by latency. This is achieved by the introduction, between a primary and a standby database in an Active Data Guard configuration, of a lightweight Oracle instance – the Far Sync instance. A Far Sync instance has a standby control file, standby redo logs, and archive redo logs – but no data files. It is deployed at a distance from the primary so that the primary can tolerate the network latency of synchronous transport to the Far Sync, which appears to the primary as a Data Guard destination. Far Sync receives redo synchronously from the primary database, and forwards redo asynchronously in real-time to its final destination, which is a full standby database.

A Far Sync configuration supports a one-step, zero data loss failover, via the same failover/switchover commands used for any Data Guard configuration. At failover, Far Sync transparently ensures that all the redo that had been synchronously transferred to the Far Sync instance is applied at the standby, thus delivering the zero-data-loss guarantee.

SQL Server has no such capability. For SQL Server, the zero data loss feature is constrained by the usual limitations of network distance and latency.

Proactive Readiness Health Checks on Standbys

In Oracle Database 12c, Data Guard Broker includes new Role Change Readiness functionality, which automatically performs comprehensive health tests to assess readiness for a failover or switchover. This leads to a reduced, more predictable Recovery Time Objective (RTO).

SQL Server does not have this comprehensive capability.

Disaster Recovery ROI – Standby Database Utilization

Data Guard Redo Apply (physical standby) has advanced capabilities that enable customers to easily utilize their standby database servers, storage, and software, for productive purposes while in standby role – hence obtaining an effective return on their DR investment. One of the premier Data Guard standby database utilization capabilities is offered through Active Data Guard – available since Oracle Database 12c Release 1. Active Data Guard enables the Data Guard physical standby database to be used as an extremely high performance real-time-synchronized replica of the production database that is simultaneously available for read-only access. This enables read-only application modules (e.g. reporting applications) to be offloaded to the Active Data Guard physical standby that in turn improves overall application throughput. In addition, Active Data Guard also allows offloading fast incremental backups to the physical standby database, improving the production server performance even further. Furthermore, as discussed earlier, Active Data Guard also serves as a real-time repository of active data blocks that can be used to automatically repair block corruptions on the primary database - completely transparent to the application.

In addition to the capabilities of Active Data Guard, a Data Guard physical standby database can be utilized as a read-write test system while in standby role with zero compromise to data protection – this capability is called Data Guard Snapshot Standby.

Microsoft SQL Server has no equivalent functionality to address most of the capabilities offered by Data Guard to productively utilize standby databases while they are in standby role.

The following sections provide further details on Data Guard advantages relative to AlwaysOn in terms of standby database utilization.

Full read consistency for queries executing at an active standby database

An Active Data Guard standby database is accessible for read-only queries and reporting while it applies changes received from primary database transactions. Active Data Guard standby databases also implement the same full read consistency model used by a primary database. An Active Data Guard standby database is able to return accurate, up-to-date results, just as if the query had been executed at the primary database. This makes an Active Data Guard standby database a reliable method for offloading workload from the primary database, substantially improving primary database performance and increasing return on investment in standby systems.

Due to Snapshot isolation, SQL Server blocks the read-only queries if there are pending transactions at the primary site. Those must be either committed or rolled back for the secondary replica to resume processing the requests. [30]

Continuous read-only access to an active standby database during replay of DDL and synchronization

DDL operations are transparently replicated to an Active Data Guard standby database without any impact to applications that may be querying the active standby while DDL log records or maintenance operations are applied.

Read-only users have continuous access to a Data Guard Active Standby Database during all phases of replication and synchronization of the standby with its primary database – whether the source of the log data is local or remote to the standby database.

Due to Snapshot isolation, SQL Server blocks the read-only queries if there are pending transactions at the primary site. Those must be either committed or rolled back for the secondary replica to resume processing the requests.

Automatic monitoring and enforcement of query SLAs on an active standby database

Beginning with Oracle Database 11g Release 2, Active Data Guard enables configurable service level agreements (SLA) that can be implemented using the session parameter, `STANDBY_MAX_DATA_DELAY`. The value for this parameter specifies a limit for the amount of time (in seconds) allowed to elapse between when changes are committed on the primary and when they can be queried on an active standby database. The Active Data Guard Standby will return an ORA-3172 error code if the limit is exceeded. Applications can respond to this error similar to a disconnect, and redirect the query to another active standby database or to the primary database to achieve the required SLA. This relieves the administrator from monitoring standby apply progress or responding to events (such as interruption in network connectivity between primary and standby database) that can impact the ability of an active standby database to be current for reporting requirements.

SQL Server does not have any equivalent functionality for an active standby database.

Enhanced reporting and global temporary table support on read-only standby

Active Data Guard in Oracle Database 12c enables richer reporting functionality on the standby, by enhancing support for DML on global temporary tables, separating temporary table undo and more flexible sequence support which grants a unique range of sequences that can be accessed on each standby.

SQL Server does not support this feature.

Fully supported with active-active database clustering

Active Data Guard is fully integrated with Oracle RAC. This means that the Active Data Guard standby can be a multi-node RAC database, with one node designated as the redo apply node, while the other nodes designated as scale-out reporting / query instances, thereby leading to very effective systems utilization across the entire Data Guard configuration.

In contrast, no read-write operations are allowed on secondary instances of SQL Server AlwaysOn FCI.

Dual-purpose standby database as test system

A single command can convert a Data Guard 11g physical standby database to an open read-write database without any impact to primary performance or data protection. This functionality is called Data Guard Snapshot Standby, enabling direct read-write access to a complete point-in time snapshot of the primary database. A Data Guard Snapshot Standby is an ideal QA system for pre-production testing or for any other activity that requires temporary read-write access to a copy of the primary database.

During operation, a Snapshot Standby continues to receive and archive, but does not apply, redo data transmitted by its primary database. Should a failover be necessary or when QA testing is complete, it is converted back into a standby database via a single command that discards all local updates made during testing. The standby is automatically resynchronized with the primary database using the redo data archived locally while it was open read-write. There is no limit to the number of times this process can be repeated.

A Snapshot Standby can also be "reset" as many times as needed by using a guaranteed restore point, to enable multiple iterations of test runs against an identical database – making this an ideal complement to Oracle Real Application Testing.

SQL Server AlwaysOn AG does not allow a secondary replica database to transparently and with a single command serve dual-purposes of being open read-write for test and other uses that require read-write access to the standby database, while also providing data and disaster protection for current transactions executing at the primary database. A SQL Server secondary replica is also unable to transition back and forth to a read-write database as many times as the user desires, while maintaining disaster protection.

10s of standbys for non-stop protection

Data Guard supports multiple standby databases with very flexible configuration options within a single Data Guard configuration. For example, a primary database may have a local synchronous standby database and a second remote asynchronous standby (up to 30 directly connected standby databases are supported with additional standby databases in a cascaded mode). Failover to any of the physical standby database results in the remaining standby database automatically recognizing the new primary database, thus providing continuous data protection throughout the failure event.

In addition to increased data protection and availability, this also makes it very easy to deploy multiple Data Guard standby databases to serve other uses (offload ad-hoc queries, reports, scale read performance, backups, perform QA testing, execute database rolling upgrades, migration, etc.). An excellent illustration of this is the use of a single Active Data Guard configuration by Apple for both data protection and to easily scale read-only performance to handle peak demands during holiday seasons [18].

AlwaysOn Availability Group supports only up to 4 standbys and can only configure 2 of them in synchronous mode of replication.

Fast recovery from human error and logical corruptions

Human error is one of the leading causes of downtime. Such errors lead to logical corruptions that can be widespread and result in significant downtime for point-in-time recovery operations. Data Guard can quickly recover from such corruptions using delayed apply – which delays the processing of redo on the standby by a configurable period of time. This provides administrators the opportunity to failover to the standby before the error is applied to the standby database, or to export still valid data from the standby and use it to surgically repair the primary database. A typical example of such an error is an incorrect batch job run on the primary database.

A compromise that accompanies using a delay is that the standby database must first apply the backlog of log data before it can assume the primary database role – increasing downtime should a failover be required. Oracle Flashback Database offers Data Guard users a second method to achieve the same level of protection by enabling the primary and standby databases to be quickly recovered to a previous point in time. Flashback Database eliminates any delay at failover time because the standby database is always up to date. Flashback Database is the recommended approach when also using an Active Data Guard standby for up-to-date queries and reporting.

AlwaysOn does not have any of the above capabilities.

Controlled automatic failover for ASYNC to comply with user configurable data loss SLA’s

Data Guard 12c extends Fast-Start Failover (automatic database failover) to support Maximum Performance mode (asynchronous redo transport) by adding a configurable data loss threshold that guarantees an automatic failover will never result in data loss greater than the desired recover point objective (RPO). Just as in an automatic failover using synchronous transport, the original primary is automatically reinstated as a standby database without requiring a restore from backup.

Users can also configure an automatic failover to occur immediately based on designated health check conditions or any desired ORA-nnnnn error. A new DBMS_DG PL/SQL packaged enables applications to also notify Data Guard to initiate an automatic failover.

AlwaysOn Availability Group does not support automatic failover for asynchronous configurations.

Interruption in asynchronous transport has no impact on primary database performance

Primary database availability and performance in a Data Guard configuration using asynchronous transport services are not impacted by an inability to transmit log records to a standby databases.

In an AlwaysOn asynchronous replication configuration, any interruption in the transmission of log data will prevent the transaction log of the primary database from being truncated. If the primary database transaction log becomes full, the primary database is unable to process any new transactions.

Pausing standby apply does not impact primary performance or data protection

Pausing the apply process on a Data Guard standby database has zero impact to primary database performance or data protection. The primary database will continue to process transactions and will transmit the redo data to the standby database where it is archived until the apply process is restarted and synchronization can resume. Since redo data is continuously transmitted whether apply is running or not, data is protected at all times.

In contrast, pausing the standby apply process in a replicating session results in the primary replica being unable to send any new log records to the secondary server. All records, therefore, remain active and accumulate in the transaction log of the primary database, preventing the transaction log from being truncated. If the database mirroring session is paused for too long, the log can become full and cause the primary database to stall. In addition, all data that is stranded in the transaction log of the primary database is unprotected by the standby, and vulnerable to data loss.

No performance impact while creating standby databases

Standby creation in a Data Guard configuration has zero impact on primary performance. Data Guard will also automatically synchronize the primary and standby databases for any transactions that occur after the time of the backup utilized to instantiate the standby database.

Creating a secondary replica minimally requires taking a full backup of the primary database and a subsequent log backup and restoring them both onto the standby server instance. While the standby is being created, a busy primary database can generate a high volume of new transaction log data. This can require additional transaction log backups to prevent the log from filling and the primary database from stalling. These additional transaction log backups must be manually copied and restored at the standby before mirroring can start. Furthermore from Microsoft documentation, *"We recommend that you configure database mirroring during off-peak hours because configuration can impact performance."* [17].

Silent corruptions due to hardware / system software faults are detected at both primary and standby

Data Guard provides industry-unique protection against silent corruptions caused by lost writes. A lost write occurs when an I/O subsystem acknowledges to Oracle that a write is complete, while in fact the write did not occur in the persistent storage. On a subsequent block read, the I/O subsystem returns the stale version of the data block, which can be used to update other blocks of the database, thereby corrupting it. Oracle provides a `DB_LOST_WRITE_PROTECT` initialization parameter which when set, will record buffer cache block reads in the redo log and use this information to detect lost writes. When a Data Guard standby database applies this redo during managed recovery, it reads the corresponding blocks and compares the system change numbers (SCNs) with the SCNs in the redo log to determine if there has been a lost write. The procedure recommended for repairing lost writes on a primary database is to failover to the physical standby and recreate the primary. If the lost write has occurred on the standby, simply recreate the standby database or the affected files.

Database Mirroring does not have the ability to detect lost writes and therefore avoid data loss and downtime that results when such corruptions are allowed to propagate unchecked.

Zero data loss for all planned role transitions using asynchronous transport

Database Mirroring high-performance mode (asynchronous transport), supports only one form of role transition: forced service with possible data loss – it does not have the ability to perform a zero data loss switchover for planned maintenance.

Data Guard has no such limitation when using asynchronous transport.

Automatically add new data files and subdirectories with no primary impact

Data Guard Data Guard automatically adds new data files to the standby database as they are added to the production database even if the data file is created in a new sub-directory. Additionally, directory structures of the standby do not need to be the same as the primary. In such cases Data Guard provides a parameter '`db_file_name_convert`' that can be updated to direct the standby where to place the data file. The transport and apply of changes to the standby are not interrupted during data file creation, no manual intervention is required, and there is no chance of stalling the primary database.

However, in an AlwaysOn availability group deployment, adding a file during a regular replication session without impacting data protection requires that the path of the file exist on both servers. If the path of a new file does not exist on the secondary replica, after the file is added on the principal database, the secondary replica goes into suspended mode and NOT SYNCHRONIZING state. Before it is possible to resume replication, the administrator must fix the problem by removing the secondary replica from the availability group, backing up the primary database logs and the filegroup containing the add-file operation and manually restoring them on the standby replica. The administrator must complete these manual tasks before the primary transaction log becomes full and stalls the primary database. [http://technet.microsoft.com/en-us/library/hh510190.aspx]

High performance log-gap resolution for best protection

Data Guard is used for databases with very high transaction rates that generate very high redo log volume [25]. Extended network outages or standby system maintenance can pause redo transport long enough to result in a large volume of accumulated log records that must be transmitted to the standby database. The accumulated log must be shipped much faster than the current rate that logs are being generated or the standby database can never catch up. Data Guard has the ability to transmit up to twenty-nine parallel streams of redo log records (either compressed or uncompressed) to quickly resynchronize the standby database, in addition to a thirtieth stream that transmits log records from current commits. Data Guard returns the configuration to a protected state much faster, and is much less likely than Database Mirroring to require a manual rebuild of the standby database following an extended outage.

When an AlwaysOn Availability Group replication session is paused or a primary database becomes disconnected from its standby, log records remain active and continue to accumulate in the transaction log of the primary database. When the session is resumed, the primary immediately begins sending the accumulated log records to the standby. Replication can only use a single thread per standby database to send log records, regardless of how large the volume of accumulated log.

Observer/Witness does not require a database instance

The Witness process in a AlwaysOn configuration is a full-fledged SQL Server instance, compared to the simple, lightweight, easy-to-install Observer process in a Data Guard Fast-Start Failover configuration.

Seamless integration with active-active clusters

Data Guard seamlessly integrates with Oracle RAC (a true active-active cluster where all RAC nodes have access to the same Oracle database – for both HA and scalability) and has the same behavior with respect to primary and standby databases at failover time regardless of configuration. This provides the following benefits:

- Data Guard users can take full advantage of the scalability benefits of Oracle RAC. Data Guard transparently merges redo data received from multiple primary RAC nodes to efficiently and correctly synchronize the standby database. A primary RAC database may have a single node standby or a RAC standby – whichever is the best solution for customer requirements.
- If a server in a primary RAC database fails, RAC quickly transitions and load balances the affected users across surviving servers in the primary cluster. Active user sessions running on the surviving servers are not impacted. This behavior is consistent whether using Data Guard automatic or manual failover. In the case of automatic failover, Data Guard is intelligent enough to know that the primary cluster is still available and unlike Database Mirroring, it does not preempt Oracle RAC from transparently executing a less-obtrusive local failover to surviving primary nodes.

In contrast, AlwaysOn Availability Group is limited to passive co-existence with Microsoft failover clusters (cold failover, active-passive clusters). This results in different behavior at failover time depending upon configuration [19]. For example, primary server failure in a Microsoft cluster (active-passive) configured with Database Mirroring using high safety mode (synchronous transport) and automatic failover will always trigger a complete failover to the standby database. While this occurs, Microsoft failover cluster will fail over to a passive server on the original primary cluster and make it a standby database for the new primary. Alternatively in a Microsoft cluster configured with Database Mirroring and manual failover, Microsoft cluster will fail over to a passive server and the primary cluster will retain its role hosting the primary database.

Flexible support for mixed primary/standby configurations

Data Guard supports Linux/Windows, 32bit/64bit, and other select mixed primary/standby combinations as described in Oracle Support Note 413484.1. Of course Data Guard also supports all operating systems supported by the Oracle Database.

AlwaysOn requires Microsoft Windows Operating system on both primary and standby databases.

Utilize standby databases, boost primary performance

Data Guard Redo Apply (physical standby) offers users two options for direct read-only access to the standby database while in standby role.

- Any physical standby database can be opened read-only to satisfy read-only reporting requirements. While in this state, redo is still sent to the standby server for protection – it is just not applied to the standby database while it is open read-only. After the reporting is complete, redo apply can be restarted with a simple command, or a mouse click. In this manner, a physical standby database can be transitioned back-and-forth between being in redo apply mode to resynchronize and then opened read-only for reporting, as many times as needed to suit specific business requirements.
- Beginning with Oracle Database 11g, an Active Data Guard physical standby can be open read-only WHILE redo apply is active. Read-only queries that execute on the standby database return real-time results, completely up-to-date with the production database.

Utilizing standby databases to offload the primary database of read-only queries and reporting will improve performance and scalability for all workload – both read-write transactions that remain on the primary database, and read-only transactions that are offloaded to the standby.

As an additional benefit, Data Guard physical or logical standby databases that are continuously open also offer an excellent way to validate the DR configuration without causing any disruption to the primary database. Oracle is always open at the standby database and you are able to query it at any time.

Microsoft SQL Server 2012 AlwaysOn AG now allows the secondary replicas to be opened for read-only for reporting and backup purposes. However, there are limitations on getting the latest changes from the primary replica. Refer to the Readable Secondary Replicas section covered later in this paper.

Microsoft does provide an additional workaround providing limited access to a standby database by using a feature called "database snapshots". A Microsoft database snapshot is a static, read-only, transaction-consistent snapshot of the standby database as it existed at the moment of the snapshot's creation. There are two significant drawbacks to this workaround. The first is that readers see stale data, frozen at the point in time the snapshot was created. The second is that a database snapshot grows over time as more and more database pages are updated. Because every snapshot on a database grows incrementally in this way, each database snapshot consumes as many resources as a normal database. According to Microsoft, an excessive number of database snapshots on a standby database can actually decrease the performance of the primary database. These limitations are showstoppers if read access is required to current information or if you are seeking a solution that has zero impact to primary database performance. For most requirements, a SQL Server standby database is an idle asset that lowers customer return on investment.

In contrast, clients can directly access a Data Guard standby database for read-only queries and reports, the standby is always up-to-date with the primary, and there is no requirement for additional storage capacity, management complexity or performance overhead that accompanies the use of Microsoft database snapshots.

Automatically enforce service level objectives for queries using an active standby database

Active Data Guard enhancements in Oracle Database 12c include the ability to configure service level agreements (SLA) for clients that query an active standby database by utilizing the session parameter, STANDBY_MAX_DATA_DELAY. The value for this parameter specifies a limit for the amount of time (in seconds) allowed to elapse between when changes are committed on the primary and when they can be queried on an active standby database. The active standby will return an ORA-3172 error code if the limit is exceeded. Applications can respond to this error similar to a disconnect, and redirect the query to another active standby database or to the primary database to achieve the required SLA. This relieves the administrator of manually monitoring standby apply progress or needing to immediately take action in response to events (such as interruption in network connectivity between primary and standby database) that can impact the ability of an active standby database to meet service level objectives for reporting accuracy.

SQL Server has no equivalent functionality.

Seamless conversion of standby to a read/write database, and back

Using Data Guard 11g and above, physical standby can be easily converted back and forth to an open read-write database without any impact to primary performance or data protection. This functionality is called a Data Guard “Snapshot Standby”, which unlike a SQL Server database snapshot, enables read-write access to the original standby database independent of the primary database applications. A Data Guard Snapshot Standby is an ideal QA system for pre-production testing or for any other activity that requires temporary read-write access to a copy of the primary database (ideally suited to complement Oracle Real Application Testing).

During operation, a Snapshot Standby continues to receive and archive, but does not apply, redo data transmitted by its primary database. For example, when QA testing is complete and the user is ready to convert the snapshot back to a physical standby, all local updates made to the snapshot during testing are automatically discarded, and the redo data archived while it was open read-write is applied to resynchronize the standby with the primary.

A physical standby database can be converted back and forth to / from a snapshot standby database via a single mouse clicks using Enterprise Manager Grid Control.

In contrast, a SQL Server standby database cannot be activated to be a read-write database and process transactions independent of the primary database while simultaneously protecting primary data, nor can a standby database opened and modified in such a manner ever resume its function as a standby database.

Offload fast incremental backups to a standby database

Backups taken at a Data Guard physical standby database can be used to restore the primary database. This offloads the backup operation from the production database, reduces resource contention on the production server, boosts performance, and enables no-downtime backup windows. Furthermore, using Oracle Recovery Manager (RMAN), the backups can occur while redo is being applied to the physical standby database to avoid any impact to failover time or data protection. Additionally, Active Data Guard enables very fast incremental backups on the standby database, avoiding full table scans by only backing up changed blocks – an essential feature when backing up very large databases.

SQL Server standby database cannot be used for such backups.

Database-integrated Load Balancing of Application Connections

Many customers have offloaded Read-only and Read-mostly workloads to their Active Data Guard Standby replicas, and Oracle GoldenGate replication also enables distributing workloads over multiple databases, both within and across datacenters. In Oracle Database 12c, Global Data Services (GDS) extends the familiar notion of Database Services to span multiple database instances over replicated Active Data Guard / GoldenGate configurations, in near and far locations, and provides RAC-like failover, service management, and service load balancing to these replicated database configurations. GDS benefits include:

- Higher Availability by supporting service failover across local and global databases.
- Better Scalability by providing load balancing across multiple databases.
- Better Manageability via centralized administration of global resources.

GDS provides very sophisticated database-integrated inter- and intra-region load balancing across replicated databases. For example, it can distribute load across a reader farm composed of standby instances, and even direct some read traffic to the primary if conditions warrant it. Data Guard performs role transitions, but GDS is aware of them and directs load as appropriate. With Active Data Guard, GDS supports:

- Service failover and load balancing across replicated databases in local and remote data centers.
- Automatic role-based services upon Data Guard role transitions.

- Load balancing for reader farms.

With GoldenGate, GDS supports failover and load balancing for local and remote data centers. When Active Data Guard and Oracle GoldenGate allow offloading production workloads to the replication assets, GDS enables better replica utilization, yielding better performance, scalability and availability.

With database-integrated Global Data Services, Oracle customers do not need to install separate hardware load balancers and traffic managers for their database infrastructure. In contrast, SQL Server has no such sophisticated capabilities.

Understanding SQL Server AlwaysOn Features

This section addresses feature details of SQL Server AlwaysOn and highlights the shortcomings.

Readable Secondary Replicas

In an AlwaysOn availability group, one or more availability replicas can be configured to allow read-only connections when running as a secondary replica. A secondary replica that allows read-only connections, a readable secondary replica, provides read-only access to all its secondary databases within the context of the availability group. This capability is similar to Oracle Active Data Guard. However, there are some restrictions.

Restrictions Related to Snapshot Isolation

All queries that run against the secondary databases are automatically mapped to snapshot isolation transaction level. There are a couple of related serious restrictions:

At the time a readable secondary replica joins the availability group, if any active transactions exist on a primary database, row versions will not be fully available immediately on the corresponding secondary database. Any active transactions that existed on the primary replica when the secondary replica was configured must be committed or rolled back, and that transaction record has to be processed on the secondary replica. Until this process completes, the transaction isolation level mapping on the secondary database is incomplete and queries are temporarily blocked. Microsoft states that "*This is needed to guarantee that row versions are available on the secondary replica before executing the query under snapshot isolation as all isolation levels are implicitly mapped to snapshot isolation.*" [16]. Basically, this means an active readable replica can't be introduced within the availability group without quiescing the primary replica.

Because read operations are mapped to snapshot isolation transaction level, the cleanup of ghost records on the primary replica can be blocked by transactions on one or more secondary replicas, e.g. a long running read-query on the secondary. Note that the ghost cleanup can also be blocked if secondary gets disconnected or when the data movement is suspended. This state also prevents log truncation. If this state persists for a longer duration, Microsoft recommends unjoining the secondary replica from the Availability Group, which is a serious downtime problem.

Read-only workloads on the secondary replicas can block incoming DDL changes that are applied through log records. Even though the read operations do not take shared locks because of row versioning, these operations take schema stability (Sch-S) locks, which can block redo operations that are applying DDL changes. In case of DDL getting blocked due to concurrent read-workload and the '*recovery interval*' threshold is crossed (note: '*recovery interval*' is a SQL server configuration option), SQL Server generates the *sqlserver.lock_redo_blocked* XEvent, which Microsoft suggests to use to kill the blocking readers [17], with absolute no regard to application uptime.

None of these restrictions are present with Active Data Guard.

Capturing Read-only Statistics

Statistics on columns of tables and indexed views are used to optimize query plans. The read-only workload on the secondary databases may need different statistics than those that are created on the primary databases. Since secondary databases are restricted to read-only access, statistics cannot be created on the secondary databases. To address this problem, the secondary replica creates and maintains temporary statistics for secondary databases in tempdb. *Note: The tempdb system database is a global resource that is available to all users connected to the instance of SQL Server and is used to hold Temporary user objects, internal objects, row versions, etc. Operations within tempdb are minimally logged. For further details, ref. [18].*

The suffix *_readonly_database_statistic* is appended to the name of temporary statistics to differentiate them from the permanent statistics that are persisted from the primary database. Because temporary statistics are stored in tempdb, a restart of the SQL Server service causes all temporary statistics to disappear.

Client Connectivity and Read-only Routing

Clients can be connected to the database of a given availability group by creating an availability group listener. An availability group listener is a server name to which clients can connect in order to access a database in a primary or secondary replica of an AlwaysOn availability group. Availability group listeners direct incoming connections to the primary replica or to a read-only secondary replica.

An availability group listener consists of a listener DNS name, listener port designation and one or more IP addresses. The availability group listener enables the client to connect to a replica without having to know the name of the physical instance of SQL Server it is connecting to.

Availability group listeners rely on Windows Server Failover Clustering (WSFC) functionality in order to redirect client connections in the event of local or remote failures of an availability group. If a primary replica is switched to being a secondary replica, the client need only reconnect to the availability group listener and SQL Server will then direct the client to the instance of SQL Server that hosts the primary replica.

Clients can also connect to readable secondary replicas using a SQL Server feature called *Read-only Routing*. Read-only routing refers to the ability of SQL Server to route incoming connections to an availability group listener to a secondary replica that is configured to allow read-only workloads. This requires the following configurations:

The *application intent* of the incoming connection needs to be set to *read-only* (for example, by using the *Application Intent=ReadOnly* keyword in the ODBC or OLEDB connection strings or connection attributes or properties).

The connection access supported on the secondary replica needs to be set to read-only.

The application intent attribute is stored in the client's session during login and the instance of SQL Server will then process this intent and determine what to do according to the configuration of the availability group and the current read-write state of the target database in the secondary replica.

To implement read-only routing, the READ_ONLY_ROUTING_LIST option must be set for each replica, as part of the PRIMARY_ROLE replica options. The READ_ONLY_ROUTING_LIST can contain one or more routing targets, and routing will take place in the order targets are specified in routing list.

As an example, let's assume the following ODBC connection string is specified:

```
Server=tcp:AGListener,1433;Database=AdventureWorks;IntegratedSecurity=SSPI;ApplicationIntent=ReadOnly
```

This implies that the client is attempting to connect to an availability group listener named AGListener on port 1433. The connection string has the ApplicationIntent property set to ReadOnly. Without this setting, the server would not have attempted a read-only routing of the connection. However, in this case, the primary database of the availability group processes the incoming read-only routing request and attempts to locate an online, read-only replica that is joined to the primary replica and is configured for read-only routing. The client receives back connection information from the primary replica server and connects to the identified read-only replica.

It is possible that multiple connections using read-only routing will not all connect to the same read-only replica. Changes in database synchronization or changes in the server's routing configuration can result in client connections to different read-only replicas. To ensure that all read-only requests connect to the same read-only replica, Microsoft recommends specifying the name of the read-only instance (instead of using the availability group listener) to the Server connection string keyword.

Read-only routing will take longer than connecting to the primary because read-only routing first connects to the primary and then looks for the best available readable secondary. Because of this, Microsoft advises to increase login timeout for client sessions.

Automatic Page Repair

Each availability replica tries to automatically recover from corrupted pages on a local database by resolving certain types of errors that prevent reading a data page. If a secondary replica cannot read a page, the replica requests a fresh copy of the page from the primary replica. If the primary replica cannot read a page, the replica broadcasts a request for a fresh copy to all the secondary replicas and gets the page from the first to respond. If this request succeeds, the unreadable page is replaced by the copy, which usually resolves the error.

The Database Mirroring feature in previous SQL Server releases has a similar feature, although that works only for a 1:1 primary-secondary association.

Backup on Secondary Replicas

For backups taken on a secondary replica, the BACKUP DATABASE Transact-SQL statement supports creating only copy-only backups of a full database, files, and filegroups. Creating incremental backups is not supported, which is a serious drawback compared to backup support on Data Guard physical standbys.

Another restriction of this feature compared to taking backups on a Data Guard physical standby is that that the backup of a secondary replica cannot be performed if the secondary replica is not able to communicate with the primary replica. Besides, the secondary replica must be synchronized or be in the process of synchronizing in order to perform the backup on the secondary replica.

Three related configuration settings are:

1. At an availability group level, one can specify whether backups should or should not run on the primary replica.
2. For each availability replica, one can specify whether it is a candidate for running backup jobs.
3. At the replica level, one can specify an ordered preference among secondary replicas for running backup jobs.

Indirect Checkpoints

Depending on when the SQL Server instance last performed a checkpoint operation, there can be a substantial amount of dirty pages in the buffer cache that need to be written to disk, which can lead to long and unpredictable failover time. SQL Server 2012 has a new capability, called Indirect Checkpoints (ref. [11]), which can be used to throttle the amount of dirty pages kept in the buffer cache. This is similar to the Fast-Start MTTR Target feature in the Oracle Database, which has been available since Oracle9*i*.

Response to SQL Server 2012 AlwaysOn Feature Set

Data Guard architecture and capabilities provide superior data protection and data availability without impacting the primary database.

SQL Server 2012 AlwaysOn solution stack is comprised of 3 distinct technologies:

- Windows Server Failover Clustering
- SQL Server Failover Cluster Instance
- SQL Server Availability Groups

An HA / DR architecture based on stitching together these individual components is not seamless, as is evident in the various restrictions Microsoft has outlined.

Besides, prior versions of SQL Server witnessed the creation of their own set of HA / DR technologies such as Database Mirroring, that were announced with a lot of fanfare. However, with this release of SQL Server, Microsoft is recommended to migrate away from those technologies.

Twelve years have gone by since SQL Server 2000 was released. Still Microsoft still not able to develop a consistent and robust set of HA / DR technologies that can be used in mission critical deployments.

Availability Groups – Only Four, or Two

A primary replica allows only up to four secondary replicas within an Availability Group (max 2 in synchronous commit mode). While this is better than the restrictive 1:1 primary-secondary model allowed in Database Mirroring, this is still lagging compared to Data Guard which allows 30 direct standby destinations, and potentially many more with cascaded destinations. This makes Active Data Guard very suitable for reader farm deployments satisfying the need for massive scale-out Internet-facing businesses, something that is not possible with Availability Groups.

Availability Groups – No Asynchronous Switchover!

Microsoft makes it a big deal about Availability Groups and says how this technology can be used for administrative tasks such as Upgrades or similar maintenance activities However, if the Availability Mode is asynchronous, the only possible role transition is Forced Failover, with Data Loss! The only way to avoid losing data for planned maintenance operations is configuring for synchronous Availability Mode, which will incur a serious impact for WAN deployments. So this basically means this Availability Group technology cannot be effectively used for most planned maintenance needs.

Half-baked Automatic Failover

SQL Server Failover Cluster Instance supports automatic failover. Availability Groups support automatic failover. So when customers try to put these together, they no longer get automatic failover! Unfortunately, when these two technologies are put together, race conditions come up and hence this configuration will no longer allow automatic failover to an availability group replica. This reinforces the premise that AlwaysOn is really based on a loose foundation of disparate technologies.

Super Long RTO

Microsoft presents the SQL Server Multi-Subnet Clustering architecture as a viable HA/DR architecture. Their articles explicitly state that when metrics such as DNS replication propagation latency and time-to-live (TTL) are accounted for, clients will take minimum 16 minutes to reconnect to the new primary instance. SQL Server has no database-integrated client notification mechanism such as Oracle Fast Application Notification (FAN). With no support for today's stringent Recovery Time Objective (RTO) demands, this technology is practically useless.

Lacking Best RPO

SQL Server allows Forced Failovers, however – in cases when the Availability Mode was asynchronous, SQL Server has no integrated method to automatically send the last bits of data from the old primary database to the new primary. Oracle Data Guard does have this support through the Flush Redo feature, which enables zero data loss and support for the best Recovery Point Objective (RPO) even in forced failover situations. Without this support, Microsoft recommends this: *As long as any of the secondary databases remains suspended, you have the opportunity to manually salvage missing committed data !!*

Readable Secondary with Restrictions

Because of the snapshot isolation transaction level for readable secondary replica, SQL Server readable secondary has restrictions that have an impact on the primary database! The cleanup of ghost records on the primary database is impacted by long running queries on the readable secondary. The readable secondary also cannot join the Availability Group if there are active transactions on the primary database. In addition, incoming DDL changes are blocked by queries on the secondary. Finally, the readable secondary allows backups to be offloaded, but these backups can only be full copy backups – incremental backups are not supported. None of these restrictions are present with Data Guard.

SQL Server Multi-Subnet Clustering – Reliance on Storage Mirroring

SQL Server Multi-Subnet Clustering architecture supports a stretch cluster model – still active-passive, but worse, it relies on third-party storage mirroring technologies to protect the database. There is no integration whatsoever between Windows Server Failover Clustering and the underlying mirroring technology. If there is instance-level failover at the cluster level, there is no coordination with possible storage-array-level failover. Customers are on their own to figure this out.

SQL Server’s High Availability Shortcomings

Any DR or HA solution must address availability of secondary replicas during network failures. It should also provide easier and convenient manageability for high availability and disaster recovery scenarios. Unfortunately, SQL Server Availability Group comes short in both these aspects.

Secondary replicas may go offline in the event of a network failure. This hugely impacts the overall availability and also the distribution of read-only queries. [Refer #31]

SQL Server architecture heavily relies on Microsoft Failover Cluster technology. Hence SQL Server DBAs must have a great understanding of Clustering technology too. This makes SQL Server high availability complex to manage and troubleshoot. [Refer #32].

Addressing Human Errors

A leading cause of data failure and application downtime is human error, be it accidental or malicious. Traditionally, while it might take minutes to damage a database, it would take hours to recover the original data. Moreover, human errors generally cannot be detected by the database, as erroneous changes are processed just like any other changes while the database remains operational. The real challenge is in identifying such errors and taking the fastest route for recovery.

As shown in the following table, Oracle provides clearly differentiating capabilities compared to SQL Server in terms of how effectively it addresses human error situations. Oracle Flashback Technologies reduce recovery time from hours to minutes, unlike traditional solutions that may take hours as they require restoring a lot or all of the database state as it was prior to the incident. Flashback Technologies are a revolution in recovery technology, because they deliver on the principle that the time to recover a database be independent of the size of the database.

TABLE 5: ADDRESSING HUMAN ERRORS – ORACLE VS. SQL SERVER

ADDRESSING HUMAN ERRORS	ORACLE	SQL SERVER
Built-in capability to mine logs and audit changes using a SQL interface	Yes	No
Built-in capability to unwind granular transactions	Yes	No
Built-in capability to ciew changes across row versions	Yes	No
Built-in ability to unwind a table to a point in time in the past	Yes	No
Built-in ability to unwind the database to a prior point in time without restoring a backup	Yes	No
Built-in capability to recover dropped objects	Yes	No
Support Recycle Bin	Yes	No
Flexible tablespace point-in-time recovery	Yes	No

The following sections provide further details on the Oracle differentiators to address human errors.

Oracle Flashback Technologies

Oracle Flashback technologies provide point-in-time viewing and quick recovery at the row, transaction, table, and database level. With Flashback, the time it takes to fix a logical error is no greater than the time it took to make the error, and it is easy to use, e.g. a single SQL command can recover the database instead of performing complex media recovery. Flashback provides fine-grained analysis and repair for localized damage, for instance, when the wrong customer order is deleted. It also allows for the correction of more widespread damage yet does it quickly to avoid long downtime, for example, when all of this month's customer orders have been deleted.

Oracle's Flashback technologies support recovery at all levels including the row, transaction, table, and the entire database [20]. SQL Server has no similar capabilities. The following sections provide further details in this regard.

Data Recovery

Oracle's Flashback Query allows an administrator or user to view the state of the data at a point in the past without requiring any structural changes to the database. This powerful feature can be used to view and reconstruct lost data that may have been deleted or changed by accident. Developers can use this feature to build self-service error correction into their applications, empowering end-users to undo their errors without delay, rather than burdening administrators with this task. Flashback Query is easy to manage, as the Oracle server automatically keeps the necessary information to reconstruct data for a configurable time in the past. In contrast, SQL Server has no ability to query data at a point in time.

To recover accidentally dropped objects, Oracle provides Flashback Drop, which accesses the Recycle Bin, a logical container for all dropped objects and their dependent objects. The recycle bin utilizes the free space in each tablespace, and purges objects on a first-in, first-out basis. When a table is dropped, it is not actually physically deleted, but simply "moved" to the recycle bin and renamed with a prefix of BIN$$. All associated table attributes are also renamed. The dropped objects are still accessible by their new name, and each user retains the same rights and privileges on them, as before. The Recycle Bin is always available by default, and no additional setup is required. SQL Server neither provides a recycle bin, nor any capability to quickly recover dropped objects.

Transaction Recovery

Oracle provides simple queries for viewing a history of changes to a set of rows and examining a transaction and all its effected changes. Flashback Versions Query allows the administrator to see a record of all changes, row version by row version, between two different times, as well as the associated metadata such as transaction id and operation type. With Flashback Transaction Query, an administrator can retrieve a listing of all operations (and corresponding undo) executed by a particular transaction. Both of these operations use the existing undo data, whose retention time can be easily configured. SQL Server has no comparable abilities.

Oracle LogMiner is a powerful audit tool that enables a DBA to find and correct unwanted changes, and can be used in conjunction with Flashback Versions Query. Its simple SQL interface allows searching by user, table, time, type of update, value in update, or any combination of these. It supports a multi-versioned dictionary that gives it the ability to track a database table even as the table goes through DDL-related structural changes. LogMiner provides SQL statements needed to undo the erroneous operation. Additionally, the Enterprise Manager (EM) interface graphically shows the change history. This is much easier and quicker than restoring a backup to perform a recovery. SQL Server does not have any integrated ability to mine logs nor undo transaction changes.

Changes by a "bad" transaction's can also be easily backed out with Flashback Transaction, accessed via PL/SQL package or EM, along with backout of changes from any of its dependent transactions (i.e. transactions that later modified the same table rows originally modified by the problem transaction). Once the backout completes, the data can be verified and the changes committed or rolled back. This operation relies on the available redo logs. SQL Server does not have single-command backout of erroneous or malicious transactions.

Point-in-Time Recovery

Oracle allows full tablespace point-in-time restore and recovery with no limit on the operations that can be backed out. SQL Server does not have a concept of restoring and recovering a tablespace (i.e. set of tables) to a prior point-in-time from a backup.

When performing a point-in-time recovery, Oracle allows querying the database without terminating recovery. This is useful to determine whether errors affect critical data or non-critical structures (such as indexes). Oracle also allows trial recovery in which recovery continues but can be backed out if an error occurs. It can also be used to undo recovery if point-in-time recovery has gone on for too long.

To quickly recover from human error at the table level, Oracle provides Flashback Table, which allows fast, online recovery for a table or set of tables, to a specified point-in-time with just a single SQL command, e.g., **`FLASHBACK TABLE orders, order_items TIMESTAMP`** *`time`*. Similar to Flashback Query, Flashback Table relies on the undo data to recover the tables. SQL Server does not have this type of fine-grained recovery.

In the case of database-wide logical corruption, a point-in-time restore and recovery may be required to get the data back to a state before the corruption occurred. Oracle circumvents the time-consuming traditional restore and recovery procedure by providing Flashback Database, in which the entire database is rewound to a specific point-in-time. Because Flashback Database operates on just the changed blocks, it can complete within seconds or minutes. By issuing the Flashback Database command, blocks are retrieved from the flashback logs, which maintain a history of changed blocks, and then archived redo logs are used to recover to the specific point-in-time. SQL Server 2012 offers a partially comparable solution to Oracle Flashback Database with Database Snapshot. However, Database Snapshots have to be created at regular intervals for the recovery point objective needed, as compared to Flashback Database, which is a *continuous* snapshot bounded only by a configurable retention period. Flashback Database allows for recovery to any point-in-time within the retention period, without the need to create and manage multiple snapshots.

With SQL Server's Database Snapshot, the user has the ability to recover at the table level. But this type of recovery requires the original data from the snapshot to be manually extracted and merged with the primary database. With Oracle Flashback Table, one SQL command performs an in-place repair of the table to any point-in-time, as bounded by the undo retention period, while the users are still able to query from the table.

Oracle vs. SQL Server – Addressing Planned Downtime

Addressing System Maintenance

As business needs change, system changes may also be required. For example, business growth often entails growth in data processing volume. This may translate into a requirement for additional processing power through hardware upgrades of disks, memory, CPUs, nodes in a cluster, or entire systems. Oracle is unique in the ability to change any system resource dynamically, as shown in the following table.

TABLE 6: ADDRESSING SYSTEM MAINTENANCE – ORACLE VS. SQL SERVER

ADDRESSING SYSTEM MAINTENANCE	ORACLE	SQL SERVER
Simple online addition of cluster nodes that requires no data redistribution	Yes	No
Automatic rebalance with online adding or dropping of disks	Yes	No
Online patching	Yes	No
Rolling database upgrades for full patch-sets and major releases	Yes	No
Extensive support to adjust memory online	Yes	No
Most configuration parameters may be modified online	Yes	No

The following sections provide further details on these capabilities provided by Oracle.

Adding a Cluster Node

With SQL Server Data partitioning in a shared-nothing environment makes adding new servers to a cluster time consuming and costly, because redistribution of partitioned data according to the new partitioning map is required. Here's what a DBA or System Administrator has to do to add a node to a SQL Server database that operates in a Federated model to support scale-out:

- Add hardware
- Configure a new partition (set partition-specific parameters, etc.)
- Restart the database (i.e. shut down and restart all nodes)
- Re-distribute the data to spread it across a larger number of partitions

Consider, on the other hand, the management tasks needed when a node is added to RAC:

- Add hardware
- Configure new instance (set instance-specific parameters, etc.)

That's it! No data re-partitioning, no offline maintenance, no database restart – just a seamless scale-out. RAC allows nodes to be added without interrupting database access.

Adding or Dropping Disks Online

With ASM, it is possible to add disks to, or drop disks from the disk group that the Oracle database is actively using, without causing any downtime to the database. ASM automatically rebalances a disk group whenever disks are added or dropped, ensuring that database files are spread evenly across all disks in a disk group. This means that administrators do not need to search for hot-spots in a disk group and manually move data around to restore a balanced I/O load. SQL Server does not have any such integrated capability – for example, it has to rely on the underlying platform support (e.g. Microsoft Windows Server –based hardware) for hot swapping of storage drives.

Online Patching

With Oracle Database 12c, it is possible to install certain one-off database patches for some selected platforms, completely online, without requiring the database instance to be shutdown, and without requiring RAC or Data Guard configurations. SQL Server does not have any similar capability. Any rolling upgrades of SQL Server – whether Service Pack upgrades, or database major release upgrades – requires the Database Mirroring feature.

Comprehensive Rolling Database Upgrades

For Oracle, Data Guard may be used for rolling database upgrades across major versions and subversions. Data Guard can also be used to minimize downtime when migrating a production database to a new system, storage or architecture. Such examples include Data Guard support for different operating system versions, mixed Linux/Windows, mixed 32bit/64bit, and other select mixed primary/standby combinations.

SQL Server requires Database Mirroring for rolling database upgrades, and similarly, Oracle requires Data Guard for rolling database upgrades. However because the SQL Server rolling database upgrade process is based on database mirroring failovers, SQL Server requires changing the operating mode from high-performance (i.e. asynchronous) to high-safety (synchronous) – to ensure zero data loss in the process, and thus is inappropriate to be used in cases where Database Mirroring is deployed over a WAN [21]:

> *Note that for high-performance mode sessions in which the mirror server is geographically distant from the principal server, a rolling upgrade might be inappropriate.*

An Oracle rolling upgrade mechanism is based on the Data Guard switchover process, which has no distance limitations, and thus is much more flexible than that of SQL Server.

Addressing Data Maintenance

As business requirements and processes change, the underlying data has to be maintained and transformed to suit the new environment, and done in such a way that there are minimal or no disruptions to the business. Maintaining, re-defining and transforming the data that supports a business is a critical activity for any DBA – this may be required unexpectedly with new business conditions, or this may even be a regularly scheduled activity. As shown in the following table, Oracle, in contrast to SQL Server, provides a suite of capabilities in this regard.

TABLE 7: ADDRESSING DATA MAINTENANCE – ORACLE VS. SQL SERVER

ADDRESSING DATA MAINTENANCE	ORACLE	SQL SERVER
Online add, drop, exchange, move partitions	Yes	No
Online reorganization of individual tables, including relocating table to a different tablespace	Yes	No
Online reorganization of individual table partitions	Yes	No
Extensive online table redefinition capabilities, including data transformations	Yes	No
Fast online add column, with default value	Yes	No
Online rename and merge columns	Yes	No
Invisible indexes	Yes	No
Online add/modify constraint, add column, index create/rebuild do not require exclusive lock	Yes	No
DDL operations wait for user-specified time, if underlying resource is busy	Yes	No

Oracle offers a wide range of online index and table reorganization operations, from the ALTER INDEX and ALTER TABLE commands to management of more complex reorganization tasks via Online Redefinition. In particular, Oracle's unique Online Redefinition capability allows one to:

- Modify the storage parameters of a table
- Move a table to a different tablespace
- Add, modify, or drop one or more columns in a table
- Add or drop partitioning support
- Change partition structure
- Change physical properties of a single table partition, including moving it to a different tablespace in the same schema
- Change physical properties of a materialized view log or an Oracle Streams Advanced Queueing queue table
- Add support for parallel queries
- Re-create a table to reduce fragmentation
- Change the organization of a normal table (heap organized) to an index-organized table, or do the reverse
- Convert a relational table into a table with object columns, or do the reverse
- Perform data transformations during online table redefinition, e.g. convert LONG data types to LOB, compute new column default values in redefined table based on original table

SQL Server cannot perform online table and partition redefinition, including even simple changes to tables, nor online add/drop/exchange/move partition operations. Additional Oracle differentiators are discussed below.

Online Data File Move and Online Partition Move

With Oracle Database 12c, a data file can be moved online while it is open and being accessed with a simple `ALTER DATABASE MOVE DATAFILE` command. Being able to move a data file online means that many maintenance operations, such as moving data to another storage device or moving databases into Oracle Automatic Storage Management (ASM), can be performed while users are accessing the system. This ensures that continuity of service and service-level agreements (SLA) on uptime can be met.

Similarly, with Oracle Database 12c, the `ALTER TABLE … MOVE PARTITION … ONLINE` statement enables one to relocate data of a partition of a partitioned table into a new segment, or optionally into a different tablespace with additional quota, or modify any of the storage attributes (e.g. compression) of the partition – all in an online manner, which enables

DML operations to run uninterrupted on the partition or subpartition that is being moved. Global indexes are also maintained during the move partition, so a manual index rebuild is no longer required.

SQL Server has none of these simple, yet sophisticated and online capabilities.

Online Application Upgrades using Editions

Oracle Database also supports online upgrade of applications with uninterrupted availability, via the use of the Edition-based Redefinition capability introduced with Oracle Database 12c. When the installation of the upgrade is complete, the pre-upgrade application and the post-upgrade application can be used at the same time. Therefore an existing session can continue to use the pre-upgrade application until its user decides to end it; and all new sessions can use the post-upgrade application. As soon as no pre-upgrade application sessions are left, it can be retired. The application as a whole enjoys hot rollover from the pre-upgrade version to the post-upgrade version. Editions-based Redefinitions works as follows:

- Code changes are installed in the privacy of a new edition.
- Data changes are made safely, by writing only to new columns or new tables not seen by the old edition. An editioning view exposes a different projection of a table into each edition to allow each to see just its own columns.
- A crossedition trigger propagates data changes made by the old edition into the new edition’s columns, or (in hot-rollover) vice-versa.

SQL Server 2012 lacks similar support for upgrading applications online.

Fast Online Add Column

With Oracle, adding new columns with DEFAULT value and NOT NULL constraint does not require the default value to be stored in all existing records. Instead, default values of columns are simply maintained in the data dictionary. This not only enables a schema modification in sub-seconds and independent of the existing data volume, it also consumes virtually no space. SQL Server does not offer online add column, as a schema lock is required, nor does it offer this fast add column optimization [22]:

> *The Database Engine uses schema modification (Sch-M) locks during a table data definition language (DDL) operation, such as adding a column or dropping a table. During the time that it is held, the Sch-M lock prevents concurrent access to the table. This means the Sch-M lock blocks all outside operations until the lock is released.*

Online No-Lock Index Creation and Rebuild

Oracle’s online index and rebuild operations do not use exclusive locks at any time during the operation. This means that ongoing DML (i.e. update, insert, delete) operations on the table work transparently and do not wait for the index operations to finish, thereby minimizing the drops and spikes in system usage that can occur with locks/waits. SQL Server’s ‘online’ index creation and rebuild, in fact, requires exclusive locks during the preparation and finish stages of the index operation, so there are two periods of time where concurrent user activity can halt [23]. Thus, SQL Server's use of the term 'online' is inaccurate.

Invisible Indexes

An Oracle invisible index is an alternative to making an index unusable or dropping it. An invisible index is maintained for any DML operation, but is not used by the optimizer unless the index is explicitly specified with a hint.

Invisible indexes have great uses in application development and testing. Applications often have to be modified without being able to bring the complete application offline. Invisible indexes enable you to leverage temporary index structures for certain operations or modules of an application without affecting the overall application. Furthermore, invisible indexes can be used to test the removal of an index without dropping it right away, thus enabling a grace period for testing in production environments. SQL Server has no such equivalent index capabilities.

DDL Wait

Oracle DDL (Data Definition Language) operations affect the logical structure of database objects and may require a lock on the underlying object, e.g. drop column, drop table. Oracle allows the user to specify a wait time for a DDL operation to complete, rather than just immediately failing if a needed lock cannot be acquired. The WAIT option allows developers to define grace periods so that DDL operations can eventually succeed, instead of raising an immediate error, and thus requiring additional application logic to handle such errors. SQL Server offers no such grace period for database structure modification operations.

Oracle High Availability Best Practices

A well-integrated set of technologies is essential in meeting the stringent high availability demands of today's complex business operations. What is also important is a set of best-practices guidelines that accompany these technologies, so that the implementation can be done in the most efficient manner.

The Oracle Maximum Availability Architecture (MAA) [24], is Oracle's best practice blueprint focused on High Availability to achieve this goal. It provides the following benefits:

- MAA reduces the implementation costs for a highly available Oracle system by providing detailed configuration guidelines. The results of performance impact studies for different configurations are highlighted to ensure that the chosen highly available architecture can continue to perform and scale accordingly to business needs.
- MAA provides best practices and recovery steps to eliminate or minimize downtime that could occur because of scheduled and unscheduled outages such as human errors, system faults and crashes, maintenance, data failures, corruptions, and disasters.

- MAA gives the ability to control the length of time to recover from an outage and the amount of acceptable data loss under disaster conditions thus allowing mean time to recovery (MTTR) to be tailored to specific business requirements.

MAA is designed, architected and validated by Oracle's High Availability Systems Engineering group, which is comprised of individuals with deep-domain design and implementation expertise in high availability best practices, as well as various Oracle and related systems technologies.

Microsoft does not offer the same level of integrated HA best practices for SQL Server.

Oracle High Availability Customers

Oracle's high availability capabilities have strong customer adoption and are a critical differentiator when the time comes for a prospective customer to choose a database technology that can support the 24x7 uptime requirements of today's businesses. A long list of customers who have implemented various Oracle high availability solutions, along with detailed implementation case studies, is available at [29].

Conclusion

Recognizing the high availability challenges every business faces, Oracle provides comprehensive, unique, powerful, and simple-to-use capabilities that protect businesses against all forms of unplanned downtime, including system faults, data corruption, disasters, and human errors. Oracle achieves this in an environment where the downtime that occurs during planned maintenance activities is also minimized.

Oracle offers a well-integrated high availability solution stack – comprised of components such as RAC, ASM, Data Guard, GoldenGate, Global Data Services, Application Continuity, RMAN, Flashback, etc. that work across multiple platforms in an integrated and efficient manner. This saves customers time, money and system/people resources – factors that are extremely critical in today's economy. Oracle has gone one step further by publishing best practice guidelines for configuring a High Availability solution through its Maximum Availability Architecture framework, and making it available for its customers. The long list of Oracle customers who have embraced its High Availability solutions is a testimonial to Oracle's unparalleled technical leadership and vision in this area.

In contrast to Oracle, SQL Server offers a basic set of high availability features and lacks the completeness and depth of High Availability functionality required by most businesses today. SQL Server continues to lag several releases behind Oracle in this regard and is not an appropriate choice for today's business applications demanding high levels of uptime.

Reference

1. Oracle Database High Availability Overview 12c (12.1) – http://docs.oracle.com/cd/E16655_01/server.121/e17601/toc.htm
2. Overview: Oracle Database High Availability – http://www.oracle.com/technology/deploy/availability
3. Oracle Database 12c documentation library http://www.oracle.com/pls/db121/portal.all_books
4. Microsoft SQL Server 2012 Books Online – http://technet.microsoft.com/en-us/library/ms130214.aspx
5. SQL Server 2008 Books Online - Federated Database Servers - http://msdn.microsoft.com/en-us/library/ms190381.aspx
6. Failover Clustering – http://www.microsoft.com/Windowsserver2008/en/us/clustering-home.aspx
7. SQL Server 2012 Books Online - Getting Started with SQL Server 2012 Failover Clustering – http://msdn.microsoft.com/en-us/library/ms189134.aspx
8. Technical Comparison of Oracle Real Application Clusters 11g and Microsoft SQL Server 2008 – http://www.oracle.com/technology/products/database/clustering/pdf/wp_oracle_msft_ss_2009.pdf
9. Oracle Secure Backup – http://www.oracle.com/technology/products/secure-backup/index.html
10. SQL Server 2012 Always On Technologies – http://download.microsoft.com/download/c/a/f/caff7135-8d80-4dad-a104-0da8558d8a0e/Availability_DataSheet.pdf
11. SQL Server 2012 Books Online - Backup Compression (SQL Server) - http://msdn.microsoft.com/en-us/library/bb964719.aspx
12. Oracle Database Backup and Recovery User's Guide 12c - Binary Compression for Backup Sets - http://docs.oracle.com/cd/E16655_01/backup.121/e17630/rcmcncpt.htm#BRADV89481
13. Oracle Database Administrator's Guide 12c - Managing Resource Allocation with Oracle Database Resource Manager - http://docs.oracle.com/cd/E16655_01/server.121/e17636/dbrm.htm#ADMIN027
14. Overview: Automatic Storage Management – http://www.oracle.com/technology/products/database/asm/index.html
15. SQL Server 2012 Books Online - Asynchronous Database Mirroring (High-Performance Mode) - http://msdn.microsoft.com/en-us/library/ms187110.aspx
16. SQL Server 2012 Books Online - Database Mirroring and Log Shipping - http://msdn.microsoft.com/en-us/library/ms187016.aspx

17. SQL Server 2012 Books Online - ALTER DATABASE Database Mirroring - http://msdn.microsoft.com/en-us/library/bb522476.aspx
18. Apple Inc, Oracle 12c Active Data Guard: High Scalability, High Availability, Real-Time Data Changes, and Reader Farm - http://www.oracle.com/technology/deploy/availability/pdf/oracle-openworld-2009/311400_01.pdf
19. SQL Server 2012 Books Online - Database Mirroring and Failover Clustering - http://technet.microsoft.com/en-us/library/ms191309.aspx
20. Overview: Oracle Flashback Technology - http://www.oracle.com/technology/deploy/availability/htdocs/Flashback_Overview.htm
21. SQL Server 2012 Books Online - How to: Install a Service Pack on a System with Minimal Downtime for Mirrored Databases - http://msdn.microsoft.com/en-us/library/bb497962.aspx
22. SQL Server 2008 Books Online - Lock Modes - http://msdn.microsoft.com/en-us/library/ms175519.aspx
23. SQL Server 2012 Books Online - How Online Index Operations Work - http://msdn.microsoft.com/en-us/library/ms191261.aspx
24. Overview: Oracle Maximum Availability Architecture (MAA) – http://www.oracle.com/technology/deploy/availability/htdocs/maa.htm
25. Oracle Customers – http://www.oracle.com/customers/index.html
26. Oracle High Availability Case Studies – http://www.oracle.com/technology/deploy/availability/htdocs/HA_CaseStudies.html
27. SQL Server 2012 AlwaysOn Availability Group Overview http://technet.microsoft.com/en-us/library/ff877884.aspx
28. SQL Server 2012 AlwaysOn Failover Cluster Instances http://technet.microsoft.com/en-us/library/ms189134.aspx
29. Oracle High Availability Customer Case Studies – http://www.oracle.com/technetwork/database/features/availability/ha-casestudies-098033.html
30. SQL Server 2012 AlwaysOn Readable Secondary Replicas http://technet.microsoft.com/en-us/library/ff878253.aspx
31. Challenges with Availability Group http://www.brentozar.com/archive/2012/09/microsoft-sql-server-alwayson-ags-at-stackoverflow/
32. Why SQL Server’s Network connection matters? http://www.brentozar.com/archive/2012/06/why-your-sql-servers-network-connection-matters/

Technical Comparison of
Oracle Database 12c
vs. Microsoft SQL Server 2012
Focus on High Availability

October 2013
Author: Oracle HA Product Management

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
oracle.com

Oracle is committed to developing practices and products that help protect the environment