

SQLskills Immersion Event

IE3: High Availability & Disaster Recovery

Module 9: AlwaysOn Availability Groups

Joe Sack
Joe@SQLskills.com



Overview

- Availability group overview
- Readable secondaries
- Application connection support
- Partial database containment
- Migrating from Database Mirroring
- Quorum model and node votes
- Asymmetric storage considerations
- Security
- Performance



2

© SQLskills, All rights reserved.
<http://www.SQLskills.com>



Availability Group Overview (1)

- **AlwaysOn is the over-arching marketing term for 2012 HA/DR**
 - AlwaysOn Availability Groups <> AlwaysOn Failover Cluster Instances
- **Availability group contains one or more user databases that fail over together**
 - Availability database is a database that belongs to an availability group
 - Primary database is the read-write copy (limit 1)
 - Secondary database is the read-only or non-readable copy (limit 4)
- **Availability replica is the SQL Server instance (standalone or SQL FCI) that hosts the availability databases**
 - Primary replica hosts the read-write databases
 - Secondary replica(s) host the secondary copies, non-writeable copies

Availability Group Overview (2)

- **Availability group listeners support automatic client redirection to the primary replica or redirection to available readable secondaries**
 - Removes the need to designate a failover partner in the client connection string
- **Secondary replicas can also be configured to permit read-only workloads, with low latency updates from the primary**
 - Offload full (copy-only) database and transaction log backup operations
- **Leverages WSFC for health detection, failover coordination, and to provide the availability group listener functionality for connectivity across any node**
 - You can use standalone instances, SQL FCIs or a mix
 - Using SQL FCIs means *no automatic failover* of the AG

Synchronous vs. Asynchronous

- **No data loss?**
 - Up to three synchronous replicas
 - Primary replica
 - Two secondaries
 - Beyond that, asynchronous
- **Some data loss allowed? No tolerance for synchronous overhead?**
 - Up to four asynchronous replicas
- **Failover mode**
 - Automatic or manual
 - If automatic, up to two replicas (including current primary)

Standalone vs. Failover Cluster Instance

- **WSFC is required, but that doesn't narrow your options:**
 - All standalone SQL Server instances on each node (non-shared storage)
 - All Failover Cluster Instances (shared storage)
 - Mix of standalone and Failover Cluster Instances
- **But just because you can...**
 - Consider the complexity of availability groups + Failover Cluster Instances
 - Can you achieve your goals using standalone instances?

Valid Usage Scenarios

- User-database high availability and disaster recovery
- Offloading read-only activity
 - But remember – there is still only *one* read-write replica
- Offloading backups
 - Transaction log backups
 - Full backups (copy only)

Cross-Database Transactions

- As with database mirroring, cross-database transactions and distributed transactions for availability group databases are 'not supported'
- For cross-database transactions, each database commits independently, even for databases in the same availability group, as a failover can happen after one database commits a transaction but the other database does not
- For distributed transactions, post-failover the new primary replica can't connect to the distributed transaction coordinator of the prior primary replica (unable to get transaction status)

Readable Secondaries

- Allow offloading of read-only workloads to secondary replicas in the environment
- Uses snapshot isolation to avoid blocking replay of transactions on the secondary
- If additional statistics are required, they are created as temporary objects in tempdb
 - If additional indexes are required, they must be created in the primary database
- Allows offloading of log backups to secondary replicas
 - All log backups form a single chain regardless of what server the backup occurred from
- FULL backups allowed with COPY_ONLY

Readable Secondary Options

- **ALLOW_CONNECTIONS = "ALL"**
 - Supports TDS clients, older clients
 - Client is responsible for ensuring replica is readable secondary
- **ALLOW_CONNECTIONS = "READ_ONLY"**
 - Only connections with "ApplicationIntent" set to "ReadOnly" are accepted
 - "Intent" means you should use read-only operations, but read/write connections are necessarily prevented
 - Will fail on the first DML or DDL operation

Read-Only Impact on Recovery Time Objective

- Page access on secondary replica can increase I/O contention with REDO thread activity, assuming collocation
- REDO thread on secondary replica can be blocked by DDL operations
 - Sch-S lock held by read-only query on a table
 - DDL transaction on the primary replica rebuilds an index for that same table
 - Sch-M lock on secondary replica
- Monitor via `lock_redo_blocked` extended event

Read-Only Impact on Data Latency

- Reporting workloads unlikely to affect the acknowledgement of synchronous commit mode
- Latency between primary and secondaries can be a few seconds to minutes depending on workload and network latency
- Data latency can still exist for synchronous mode, as the REDO thread can fall behind
- No data latency for reporting allowed?
 - Run on the primary replica

Read-Only Snapshot Isolation (1)

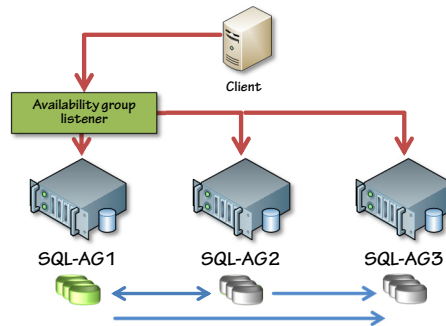
- Existing rows are not modified with 14-byte overhead until existing row is updated or deleted
- 14-byte overhead included when a new row is added
- Watch out for increased page splits on the primary

Read-Only Snapshot Isolation (2)

RCSI / SI	Read-Only	14-byte primary	14-byte secondary	Row version primary tempdb	Row version secondary tempdb
No	No	No	No	No	No
No	Yes	Yes	Yes	No	Yes
Yes	No	Yes	Yes	Yes	No
Yes	Yes	Yes	Yes	Yes	Yes

Application Connection Failover

- Clients connect using the availability group listener virtual network name
- Can specify the Application Intent=ReadOnly parameter
- During a failover the ReadOnly connections can be automatically rerouted by the Listener to a new node if the current node becomes the primary



Read-Only Routing

Prerequisites

1. Must connect using an availability group listener
2. One or more replicas need to be read-only secondaries

Read-only routing setup

1. Connect to the primary replica
2. Configure read-only for the secondary role
3. Configure a READ_ONLY_ROUTING_URL for each replica
4. Designate a READ_ONLY_ROUTING_LIST for each replica

Leveraging read-only routing

- Connect via TCP with the ReadOnly application intent, referencing the listener and a database within the availability group

Read-Only Routing Credentials

- Don't forget about the required logins on the secondary replicas
- Options:
 - Partially contained databases
 - CREATE LOGIN with SID option
 - KB 918992 sp_help_revlogin to transfer logins (applies to 2012)

Partial Database Containment

- Contained Database Authentication stores credentials inside the database removing the need to manually manage logins across environments or availability group nodes
 - Requires enabling the 'contained database authentication' sp_configure option for the instance
 - Individual databases can be converted to a contained database using ALTER DATABASE <database_name> SET CONTAINMENT = PARTIAL
 - Requires a lock
- Existing SQL Server Logins can be converted to a contained database user with the *sp_migrate_user_to_contained* system stored procedure
- New contained database users can be created using the new DDL feature CREATE USER <username> WITH PASSWORD = 'password'

Partial Database Containment

- **Temporary tables**
 - Uses the collation of the contained database instead of the tempdb collation
 - Cannot contain named constraints, or refer to user-defined types, XML schema collections, or user-defined functions
- **Instance-level features like replication, change data capture, or change tracking can not be used**
- **Feature usage that crosses the application boundary can be tracked with sys.dm_db_uncontained_entities or the database_uncontained usage Extended Event**
- **Any Dynamic SQL used will be flagged as uncontained usage since its actual containment cannot be determined until run time**

Non-contained Objects

- **Similar to database mirroring, you'll need to consider non-contained object synchronization:**
 - Logins (transfer via sp_help_revlogin)
 - SQL Server Agent jobs
 - Custom execution logic to execute as appropriate
 - Linked servers

Backup Considerations

- Do you want to offload backups to your secondary replicas?
- Even if you do not, how will you handle a failover in your maintenance jobs?
- Your maintenance must be “aware”
 - Ola’s scripts handle this (<http://bit.ly/3Jf3Go>)
 - Maintenance plans do too
 - `sys.fn_hadr_backup_is_preferred_replica`
- Availability group configurations
 - Automated backup preference
 - Backup priority

Multi-Subnet Connection Support

- SQL Server 2012 supported on Windows Server Failover Clusters with nodes that cross-subnets
- Microsoft recommends using the client `MultiSubnetFailover` attribute for both single and multi-subnet topologies that reference the availability group listener name
- Opens up TCP sockets for availability group listener IP addresses in parallel
- Older client libraries?
 - Recommendation is to increase the client login timeout
 - Consider adjusting the `HostRecordTTL` value to a lower value

Monitoring

- **Requires a combination of data sources and tools:**
 - AG Dashboard in SQL Server Management Studio
 - Object Explorer state information
 - Performance counters
 - Catalog and Dynamic Management Views
 - Extended Events session that tracks recent availability group critical activity

EMU-AG1: hosted by EMU-SQL1 (Replica role: Primary)				
Availability group state: Healthy				
Primary instance: EMU-SQL1				
Failover mode: Automatic				
Cluster state: EMU-AGClstr (Normal Quorum)				
Availability replica:				
Name	Role	Failover Mode	Synchronization State	Issues
EMU-SQL1	Primary	Automatic	Synchronized	
EMU-SQL2	Secon...	Automatic	Synchronized	
EMU-SQL3	Secon...	Manual	Synchronizing	

SQL Server: Database Replica Counters (1)

- **Log Bytes Received/sec (Secondary)**
 - Amount of log records received by the secondary replica for the database in the last second.
- **Log remaining for undo (Secondary)**
 - The amount of log in kilobytes remaining to complete the undo phase
- **Log Send Queue (Secondary)**
 - Amount of log records in the log files of the primary database, in kilobytes, that has not yet been sent to the secondary replica
 - Value is sent to the secondary replica from the primary replica
 - Does not include FILESTREAM files that are sent to a secondary
- **File Bytes Received/sec**
 - Amount of FILESTREAM data received by the secondary replica for the secondary database in the last second

SQL Server: Database Replica Counters (2)

- **Recovery Queue (Secondary)**
 - Amount of log records in the log files of the secondary replica that has not yet been redone.
- **Redo Bytes Remaining (Secondary)**
 - The amount of log in kilobytes remaining to be redone to finish the reverting phase
- **Redone Bytes/sec (Secondary)**
 - Amount of log records redone on the secondary database in the last second
- **Total Log requiring undo (Secondary)**
 - Total kilobytes of log that must be undone
- **Transaction Delay (Primary)**
 - Delay in waiting for unterminated commit acknowledgement, in milliseconds

SQL Server: Availability Replica

- This counter object applies to primary and secondary replicas
- **Bytes Received from Replica/sec**
 - Number of bytes received from the availability replica per second
 - Will generate traffic even for quiesced databases
- **Bytes Sent to Replica/sec**
 - Number of bytes sent to the remote availability replica per second
- **Flow Control Time (ms/sec)**
 - Time in milliseconds that log stream messages waited for send flow control, in the last second
- **Flow Control/sec**
 - Number of times flow-control initiated in the last second
 - Flow Control Time (ms/sec) divided by Flow Control/sec is the average time per wait

DMVs (1)

- **sys.dm_hadr_auto_page_repair**
 - Returns a row for every automatic page-repair attempt on any availability database on an availability replica that is hosted for any availability group by the server instance
- **sys.dm_hadr_availability_group_states**
 - Returns a row for each availability group that possesses an availability replica on the local instance of SQL Server and shows health descriptions
- **sys.dm_hadr_availability_replica_states**
 - Returns state information each local availability replica and a row for each remote availability replica in the same availability group as a local replica

DMVs (2)

- **sys.dm_hadr_cluster**
 - Shows WSFC cluster name, quorum type and quorum state
- **sys.dm_hadr_cluster_members**
 - Shows WSFC nodes, their current state and quorum vote
- **sys.dm_hadr_cluster_networks**
 - Networks associated with the WSFC nodes (IP address, subnet information)
- **sys.dm_hadr_database_replica_states**
 - Database replica state information, last hardened LSN and time, last redone LSN and time, log send queue size, log send rate, redo queue size, redo rate, last commit LSN, last commit time

Migrating from Database Mirroring

- Non-SQL Server staff must be involved as there are WSFC, networking, and storage considerations
- Relying on transparent client redirect?
 - Once you move past a second replica, you'll need to move to an AG Listener or other approach
- Existing database mirroring sessions?
 - SQL Server instances need to exist on nodes of the same WSFC
 - New considerations around WSFC Quorum Mode and Node Votes
 - Enable AlwaysOn HA for each SQL Server service (requires restart)
 - Remove database mirroring (SET PARTNER OFF) on principal
 - Temporarily stop log backup jobs on principal database
 - Create AG on principal instance and the existing "mirror" recovering database(s) can be "joined only" to w/o having to restore again
 - Create AG Listener

Cross-Cluster Migration of AGs

- Windows Server Failover Clustering (WSFC) does *not* support operating system rolling upgrades
- SQL Server 2012 SP1 introduces support for cross-cluster migration of availability groups for deployments to a new WSFC cluster
- A cross-cluster migration moves one availability group or a batch of availability groups to the new, destination WSFC cluster with minimal downtime
- Path
 - Source: Windows Server 2008 or 2008 R2 + SQL Server 2012 RTM or SP1
 - Destination: Windows Server 2012 + SQL Server 2012 SP1
- See "Cross-cluster Migration of AlwaysOn Availability Groups for Operating System Upgrades" <http://bit.ly/10cFqPU>

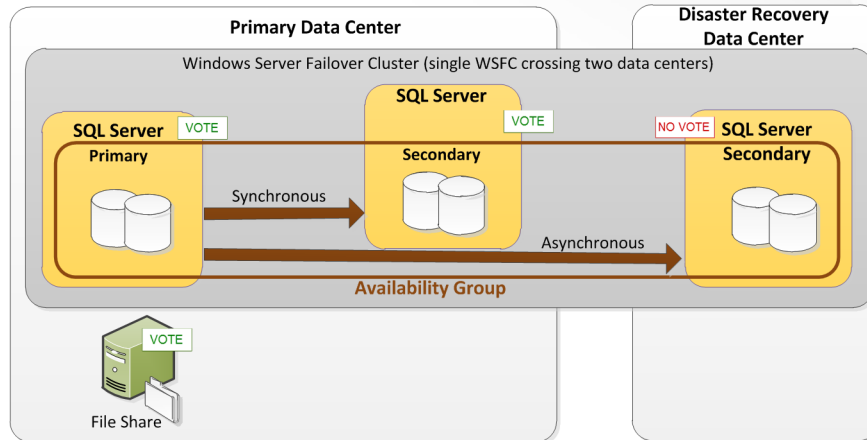
Quorum Model

- **Quorum model is managed via the WSFC**
 - Number of 'elements' needed to keep the WSFC running, and also avoid 'split brain' by ensuring majority of elements are visible
- **Must consider non-shared vs. shared-storage models**
 - Node and Disk Majority (shared disk)
 - No Majority: Disk Only (shared disk)
 - Node Majority
 - Node and File Share Majority
- **Odd vs. even number of nodes drives quorum decisions**
 - Node Majority
 - Node and File Share Majority

Node Votes

- **By default every node has a 'vote' in quorum, but this may not be appropriate for your topology**
 - Example: you may only want nodes in the primary data centers to have votes, and then no votes in the disaster recovery data center
- **Windows Server hotfix KB 2494036 allows assignment of votes via NodeWeight property**
 - Can configure PowerShell cmdlet or Cluster.exe
- **Windows Server 2012 offers:**
 - NodeWeight handling by default
 - Dynamic Quorum

Node Votes Example

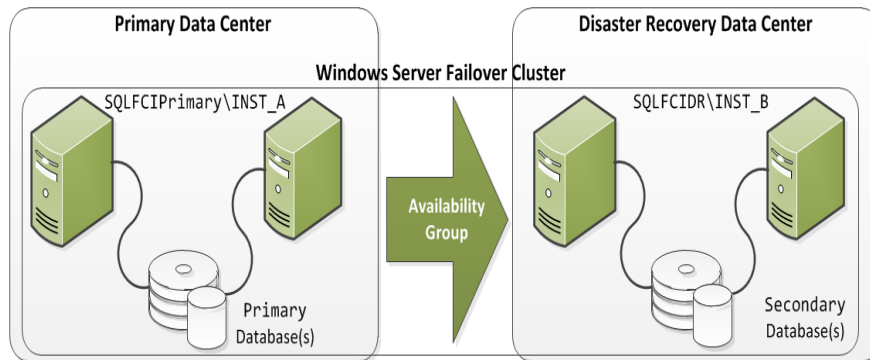


Source: "Building a High Availability and Disaster Recovery Solution using AlwaysOn Availability Groups (Joe Sack, Sanjay Mishra), <http://bit.ly/L9RbLg>

Asymmetric Storage

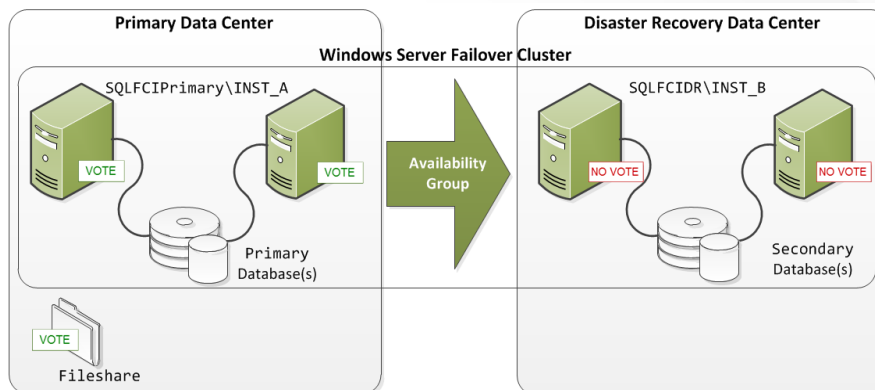
- Consider a scenario where you have an AG that has replicas on two SQL FCIs across a multi-site WSFC...
- Shared storage on the primary data center may not be visible to storage on the second data center
- Functionality to share storage between just a subset of nodes is called **asymmetric storage**
 - Symmetric storage used to be required
- Asymmetric storage supported in Windows Server 2008 R2 SP1 and Windows Server 2008 via hot-fix (KB 976097)

SQL FCI for HA and AG for DR



Source: "AlwaysOn Architecture Guide: Building a High Availability and Disaster Recovery Solution by Using Failover Cluster Instances and Availability Groups (Joe Sack, Sanjay Mishra), <http://bit.ly/Rjq154>

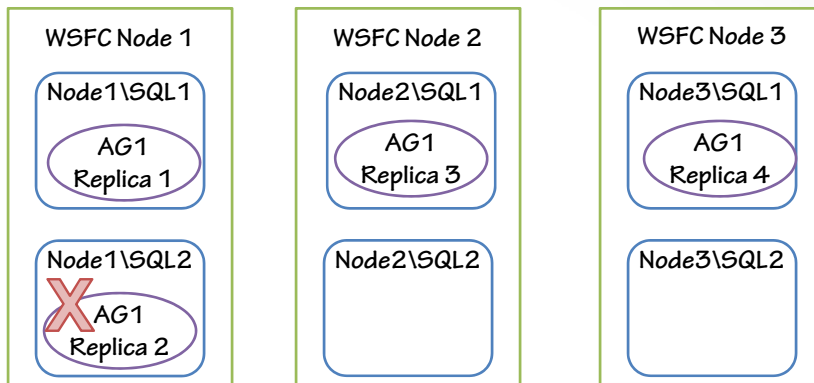
SQL FCI + AG Vote Assignments



Source: "AlwaysOn Architecture Guide: Building a High Availability and Disaster Recovery Solution by Using Failover Cluster Instances and Availability Groups (Joe Sack, Sanjay Mishra), <http://bit.ly/Rjq154>

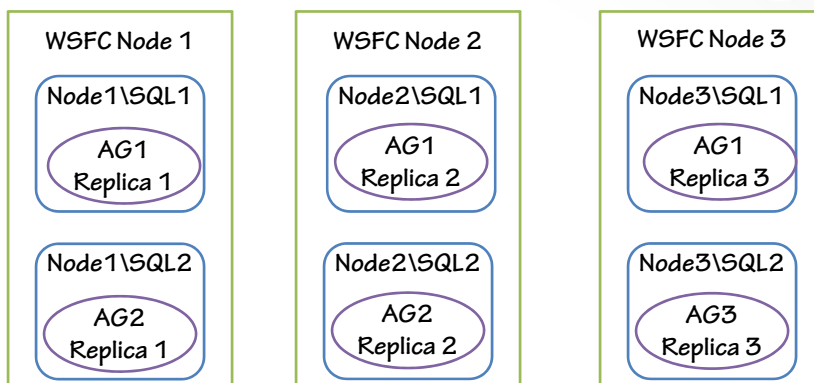
One WSFC Node per Availability Replica

A single WSFC node can only host *one* availability replica per availability group



Instances and Replicas

On one WSFC node, greater than one or more instances *can* host *different* availability group replicas



Using Failover Cluster Manager (1)

- Failover Cluster Manager (FCM) will let you do bad things to your availability group resources
- *Don't use FCM to change the following availability group and availability group listener resource settings:*
 - Preferred owners
 - Possible owners
- *Don't use FCM to fail over the availability group or availability group listener*
- *Don't use FCM to add or remove resources contained within the availability group and availability group listener resource groups*

Using Failover Cluster Manager (2)

- For Failover Cluster Instances, you may need to configure possible node owners for asymmetric storage purposes, removing those nodes that don't have access to the shared storage
 - FCI network name, IP address, SQL Server service, Agent service, other FCI resources
 - Not configuring this will cause the availability group creation to fail as the nodes can be possible owners to both replicas ("crossing streams")

Security Considerations

- **Dependent on Kerberos?**
 - Use the same service account across replicas
 - Domain admin will need to manually register a Service Principal Name (SPN) for the SQL Server service account for the virtual network name (VNN) of the availability group listener
 - Changing the account requires a manual SPN re-register
- **Deploying availability groups without Active Directory permissions?**
 - Pre-stage the Cluster Name Object

Performance Considerations (1)

- **You can have hardware disparity across nodes – but should you?**
 - I/O path and disk subsystem
 - Cores
 - Memory
- **Consider dedicated NICs for AG traffic**
 - See “Configuring a Dedicated Network for Availability Group Communication”, <http://bit.ly/STMCMQ>
 - Even on latent availability groups, you’ll see Bytes Received from Replica/sec performance counter generate persistent traffic *per database*

Performance Considerations (2)

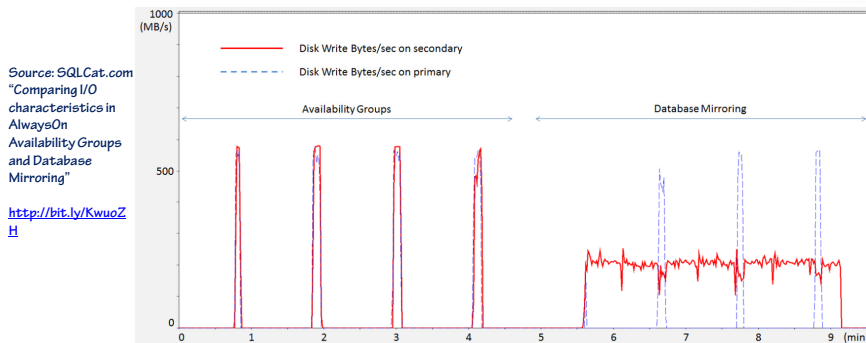
- **Failover client connectivity**
 - Newer providers .NET 4.0, SNAC can cache IPs and don't have to wait for DNS
 - MultiSubnetFailover = True
 - Older clients, consider modifying HostRecordTTL value (reducing to 5 minutes, for example) and increasing connection timeout

Worker Pool

- **Maximum threads used by availability groups is configured max worker threads minus 40**
 - Availability replicas share a single thread pool
 - Idle threads get released to the thread pool
- **Thread distribution**
 - 1 log capture thread per primary database on primary replica
 - 1 log send thread for each secondary database
 - 1 redo thread per secondary replica
 - Secondary replica backup holds a thread on the primary replica for duration of the operation

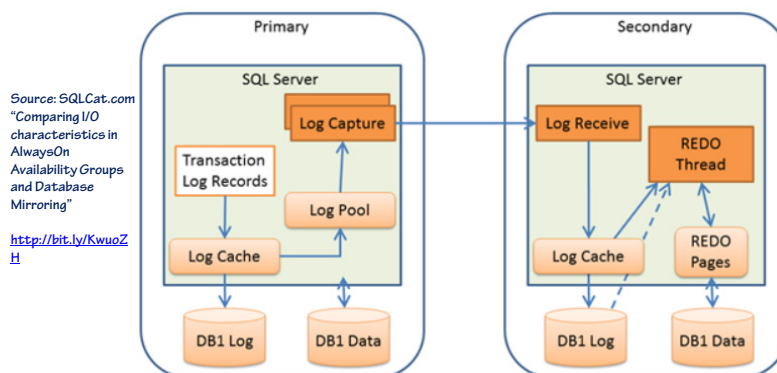
I/O Redo Improvements

- Database mirroring continually flushes dirty pages
 - Facilitates faster 'restoring' to 'online' state
- Availability groups flush on checkpoint or buffer pool pressure
 - Flush is not required on a role change given "online" state (readable or not)

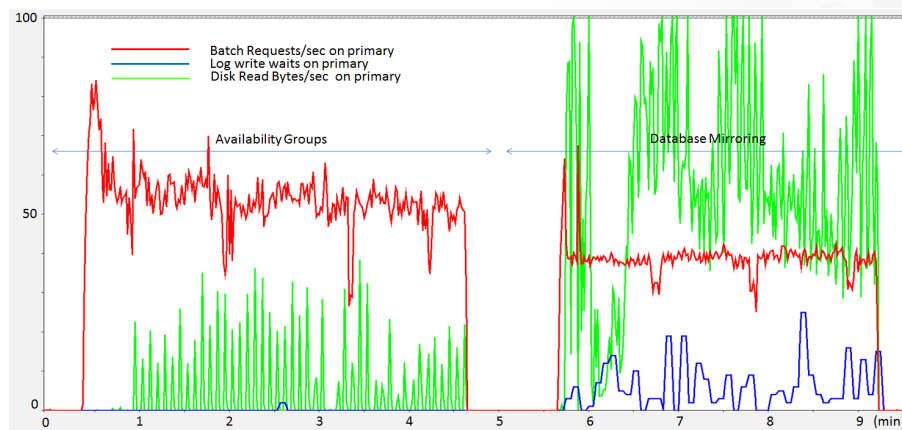


Log Pool Cache Mechanism (1)

- New log pool cache mechanism for log record consumers
- If an unsent log is not in the cache, threads have to read from the file
- The log pool is a dynamic cache that can grow if no memory pressure



Log Pool Cache Mechanism (2)



Source: SQLCat.com "Comparing I/O characteristics in AlwaysOn Availability Groups and Database Mirroring"
<http://bit.ly/KwuoZH>

Collaboration is a Requirement

- **Not just for initial configuration, but for high availability and disaster recovery scenarios**
 - DBA, Windows Server, network, storage
 - More involved than database mirroring
- **Disaster-recovery scenario definitely requires a run-book and a detailed assignment of duties, for example, on bringing up service at the disaster recovery data center:**
 - Confirm state of primary data center (DBA, Windows, network)
 - Force quorum on DR site node (Windows)
 - Force AG group to DR site (DBA)
 - Add votes to DR nodes (Windows)

Availability Group Resources

- **Whitepapers**

- AlwaysOn Architecture Guide: Building a High Availability and Disaster Recovery Solution by Using AlwaysOn Availability Groups
 - <http://msdn.microsoft.com/en-us/library/jj191711>
- AlwaysOn Architecture Guide: Building a High Availability and Disaster Recovery Solution by Using Failover Cluster Instances and Availability Groups
 - <http://msdn.microsoft.com/en-us/library/jj215886>
- AlwaysOn Solution Guide: Offloading Read-Only Workloads to Secondary Replicas
 - <http://msdn.microsoft.com/en-us/library/jj542414>

Review

- Availability group overview
- Readable secondaries
- Application connection support
- Partial database containment
- Migrating from Database Mirroring
- Quorum model and node votes
- Asymmetric storage considerations
- Security
- Performance

Questions?



Reference Slides

Availability Group Reference Slides

Hotfix: Slow Failover when no Router Exists

- Windows Server 2008 and R2:
 - <http://support.microsoft.com/kb/2582281>
- No router exists between the WSFC and application server
- A failover operation occurs and the network-related resources failover slowly
 - TCP/IP stack of the application server ignores Address Resource Protocol requests

Hotfix: IPsec Delays

- KB 980915 applies to Windows Server 2008 SP2 & Windows Server 2008 R2
 - <http://support.microsoft.com/kb/980915>
- Long delay (2-3 minutes) for a client IPsec connection to an availability group listener
- IPsec isn't optimally designed for reconnection scenarios
- Requires hotfix installation and registry changes

Hotfix: Network Bandwidth Efficiency

- KB 2616514 applies to Windows Server 2008 SP2 & Windows Server 2008 R2
 - <http://support.microsoft.com/kb/2616514>
- Without the hotfix, the cluster service will send unnecessary registry notifications across nodes
- Can reduce network bandwidth, impacting availability groups

Hotfix: False Warnings for In-Use Storage

- KB 2531907 applies to Windows Server 2008 R2 SP1
 - <http://support.microsoft.com/kb/2531907>
- **Scenario:**
 - You're using cluster disks configured in groups outside Available Storage group or using them for Clustered Shared Volumes
 - Disks are online during a "Validate SCSI Device Vital Product Data (VPD)" test or List Potential Cluster Disks storage validation
 - You see the error "Failed to get SCSI page 83h VPD descriptors for cluster disk <number> from <node name> status 2" or "Disk with identifier <value> has a Persistent Reservation on it. The disk might be part of some other cluster. Removing the disk from validation set."
- **Hotfix eliminates the warnings**

Hotfix: Faster Failover to Replicas

- KB 2687741 applies to Windows Server 2008 R2
 - <http://support.microsoft.com/KB/2687741>
- Addresses symptoms of slow failover time to another availability group replica
 - Issue due to inter-node communication issues

Hotfix: IPv6 Failover Latency

- Windows Server 2008 SP2
 - <http://support.microsoft.com/kb/2578103>
- Windows Server 2008 R2 or R2 SP1
 - <http://support.microsoft.com/kb/2578113>
- An IPv6 IP address fails over to another WSFC node
- The cluster service takes 30 seconds to fail over the address, delaying client connections
- Issue occurs due to Cluster Resource DLL (ClusRes.dll) module only sending internode discovery messages for IP version 4 (IPv4) IP addresses, and not for IPv6 IP addresses

Hotfix: Double-Free on Clusapi.dll

- SQL Server 2012 on a Windows Server 2008 R2
- When a replica goes offline, the SQL Server service crashes
 - “Double-free” memory de-allocation for Clusapi.dll
- Windows Server 2008 R2 Service Pack 1 (SP1):
 - <http://support.microsoft.com/kb/2777201>

Hotfix: Configuring Optimal WSFC Quorum

- KB 2494036 applies to Windows Server 2008 SP2 & Windows Server 2008 R2
 - <http://support.microsoft.com/kb/2494036>
- Enables you to configure which nodes have a “vote”

Hotfix: Support Asymmetric Storage

- KB 976097 applies to Windows Server 2008 SP2 & Windows Server 2008 R2
 - <http://support.microsoft.com/kb/976097>
- Applies to Failover Cluster Instances supporting availability groups
- Enabled Failover Cluster Management MMC support of asymmetric storage
 - Shared storage only available on a subset of WSFC nodes