

SQLskills Immersion Event

IE3: High Availability & Disaster Recovery

Module 6: Database Mirroring

Joe Sack
Joe@SQLskills.com



Overview

- Mirroring configuration
- Failure types and speed
- Failure scenarios
- Client redirection
- Monitoring
- Troubleshooting



2

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



Database Mirroring

- **Communicates through a dedicated TCP endpoints and continuously sends log records from the principal to the mirror (copy) database**
- **Pros**
 - Provides zero to minimal data loss through a configured database mirroring partnership, which includes a copy of the database, providing redundancy at the database level
 - Removes single point of failure when compared to failover clustering
 - No hardware dependencies *
 - Transparent client redirect for client connection management
- **Cons**
 - Performance may be impacted in synchronous (no data loss) configurations but data loss is possible in high performance configurations (again, the key trade-off in any DBM solution)
 - On the deprecation path as of SQL Server 2012

Mirroring Components

- **Principal database/server**
 - The principal server is the server which clients connect to and perform their updates and reporting
- **Mirror database/server**
 - The mirror server is performing the same changes on the mirrored database
- **Witness server (optional, lightweight, even SQL Express)**
 - The witness server monitors status of the principal and mirror servers
 - The witness does NOT trigger the failover, just helps provide 'quorum'
- **Quorum**
 - In the event of the principal becoming unavailable, the mirror can only failover if it can still see the witness, and the witness agrees it cannot see the principal
 - If the mirror fails, principal can only continue if it still sees the witness

Database Mirroring Configurations

- **High Availability: synchronous mirroring with a witness**
 - Automatic detection/failover
 - No data loss
 - Also called SAFETY FULL
- **High Protection: synchronous mirroring without a witness**
 - Manual failover
 - No data loss
 - Also called SAFETY FULL
- **High Performance: asynchronous mirroring**
 - Manual failover
 - Some data loss possible
 - Also called SAFETY OFF
 - Enterprise Edition only

Common Misconceptions

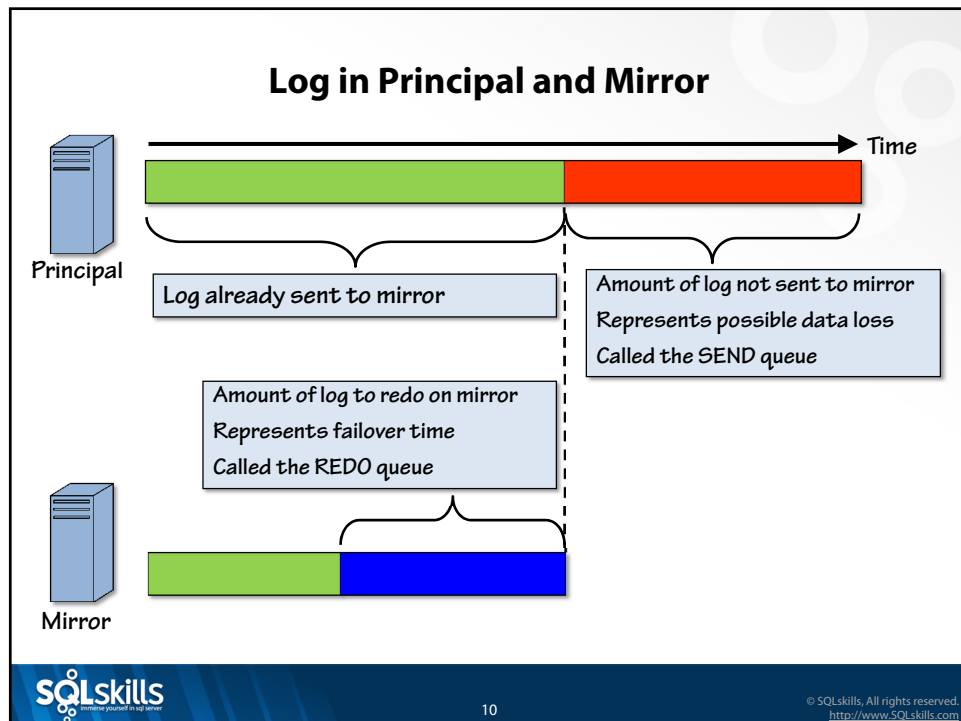
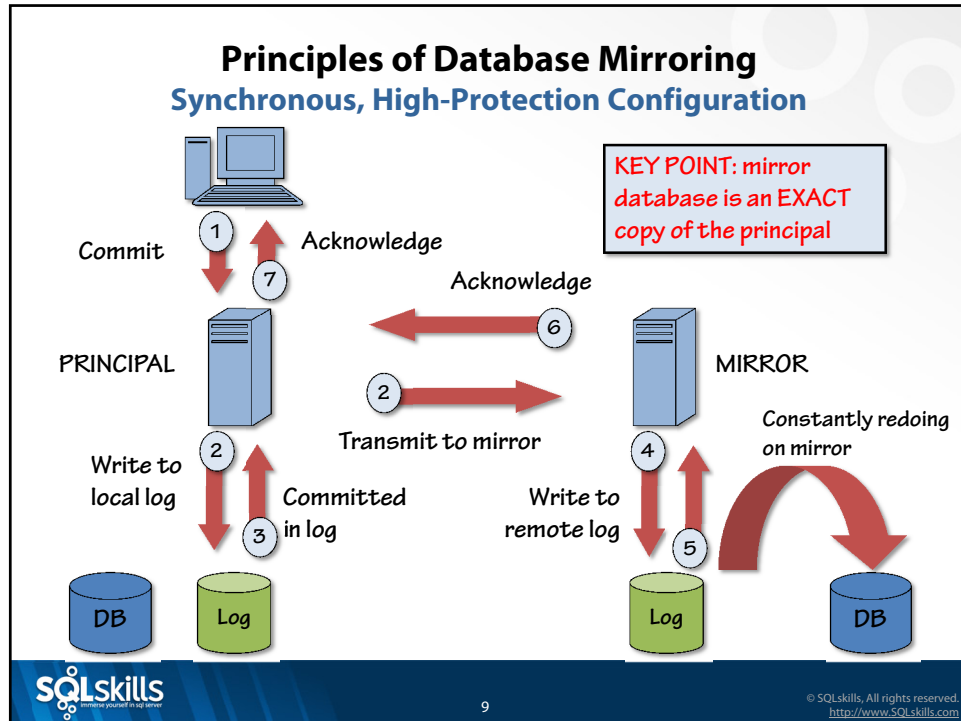
- **Can you have more than one mirror?**
 - No
- **Can you access the mirror for reads?**
 - Kind of – with a database snapshot, but backup not possible
- **Can you access the mirror for writes?**
 - No
- **How many databases can I mirror per instance?**
 - It depends...usually no more than 10 on 32-bit
 - Transaction log generation rate - workload
 - Threads
 - One global for service broker communication (principal & mirror)
 - One per mirrored database for event processing (principal & mirror)
 - One per mirrored database for async operations (principal & mirror)
 - One global redo manager thread (mirror)
 - One redo thread per mirrored database (mirror) on Standard Edition and one per four cores on Enterprise Edition

Synchronous Mode

- Provides the most data protection
- Transactions cannot commit on the principal until the log has been written to the mirror's log
 - Note: But the log doesn't have to be recovered, just written
- Failover can be either automatic or manual
 - Based on achieving quorum and the existence of a witness server
 - With a witness: high-availability mode
 - Without a witness: high-protection mode
- Not achieved until the mirror database has caught up (in SYNCHRONIZED state) with the principal database

Asynchronous Mode

- Allows for faster processing on the principal server, but at the cost of not having an up-to-date copy of the database on the mirror server
- The principal server commits each new log record locally just before sending the record to the mirror server
- The mirror server continuously applies all the outstanding log records to the mirror database in an effort to catch up to the principal database
- Only available on Enterprise Edition



Requirements for Setup

- Both the principal and mirror servers must have SQL Server 2005+ installed and have enough space for the database
 - For automatic failover, the witness server also must have SQL Server 2005+ installed
- The principal database must use the FULL recovery model
 - Implications for index maintenance
- The mirror database must be “prepared”
- Pick TCP ports for servers involved and open them in firewalls
- Make sure logins have connection permission to endpoints
- Choose encryption requirements
 - BOL: Database Mirroring Transport Security
 - <http://msdn.microsoft.com/en-us/library/ms186360.aspx>

Preparing the Mirror Database

- Back up the principal database
- Restore full backup onto the mirror server using
 - RESTORE DATABASE [dbname] WITH NORECOVERY
- Restore the latest logs to bring the mirror to current
 - RESTORE LOG [dbname] WITH NORECOVERY
- At least one log backup must be restored
- All versions require that no further log backups can be taken on the principal until mirroring is enabled
- All file paths must exist on the mirror, otherwise must use WITH MOVE on the initial restore
- Setup for mirroring

Demo

Database mirroring setup and troubleshooting

Pausing and Resuming

- **You can manually pause a mirroring session (from either partner)**
 - ALTER DATABASE [dbname] SET PARTNER SUSPEND
- **Why?**
 - For instance if you're running SAFETY FULL and there's a workload spike on the principal which overloads the network and really slows down transaction commits
 - Good example is bulk loading using SSIS – no longer bulk logged because of full recovery model
 - Pause for a while...
- **Resume when load is lower**
 - ALTER DATABASE [dbname] SET PARTNER RESUME
 - Until session is SYNCHRONIZED, transactions on principal in SAFETY FULL do NOT have to wait for log to be sent to mirror

Partner Timeout

- The mirroring partners ping each other once per second using a TCP ping
- The partner timeout is the number of unacknowledged pings before the mirror declares the principal is unavailable
- Default is 10 pings (i.e. 10 seconds)
- Can be adjusted up or down
 - Minimum is 5 seconds (even if you set it lower)
 - No maximum
- For flaky network, or when combining with failover clustering, increase the partner timeout (more in later slides)
 - ALTER DATABASE [dbname] SET PARTNER TIMEOUT X

Failure Detection

- Two types of failures:
 - SQL Server
 - Outside SQL Server:
 - Operating system
 - Network errors
 - I/O errors
 - Process errors
- Failover speed determined by:
 - Failure type
 - REDO queue size on the mirror

Examples of Failures: Fast

- **SQL Server instance crashes**
 - Endpoint closes port quickly
 - Network retry from partner quickly fails
 - OS says that the port is closed
 - Fast failure!
 - Failover begins in seconds

Examples of Failures: Not as Fast

- **Catastrophic server failure**
 - E.g. power supply fails
 - Network retry from partner waits for timeout
 - SQL Server ping will most likely fail first
 - Failover begins in 10 seconds
 - Or whatever the partner timeout value is set to

Examples of Failures: Slower...

- **Someone pulls the log drive on principal**
 - Pending I/Os to the log drive queue up
 - SQL Server pings are working fine
 - After 20 seconds, SQL Server issues I/O warning
 - After 40 seconds, SQL Server declares I/O failure
 - Failover begins 40 seconds after log drive is pulled

Examples of Failures: Either No Failover or Fast Failover

- **Database page fails page checksum validation during read**
 - Client connection is broken
 - Transaction rolls back automatically
 - No failover
- **Database page fails page checksum validation during rollback**
 - Now the database is inconsistent
 - Database goes SUSPECT
 - Fast failover!!!

Failing Over

- Once failure is detected, failover depends on configuration and state
- If in **SAFETY OFF**:
 - ALTER DATABASE [dbname] SET PARTNER
FORCE_SERVICE_ALLOW_DATA_LOSS
- If in **SAFETY FULL and SYNCHRONIZED**:
 - If the principal is still available and connected:
 - On principal: ALTER DATABASE [dbname] SET PARTNER FAILOVER
 - If the principal is not available and there's a witness:
 - Failover should be automatic
 - If the principal is not available and failover is not automatic:
 - On mirror: ALTER DATABASE [dbname] SET PARTNER
FORCE_SERVICE_ALLOW_DATA_LOSS
- If in **SAFETY FULL and not SYNCHRONIZED**:
 - On mirror: ALTER DATABASE [dbname] SET PARTNER
FORCE_SERVICE_ALLOW_DATA_LOSS

Forcing Service

- Be careful!
- When the principal comes back online:
 - If it cannot see the new principal (old mirror) and there is no witness, it will resume the principal role and accept connections -> split brain!
 - If it can see the new principal, it will become the mirror and be suspended
 - Resuming will cause the old principal to roll back to the same point as the mirror, even losing previously committed transactions
 - Salvage changes first if you need to

Mirror to Principal Switch

- Clients are disconnected
- Mirror records the failover LSN
- Mirror finished performing REDO
 - Potentially in parallel
 - Represents largest portion of downtime during failover
- (Mirror comes online on Enterprise Edition)
- Mirror performs UNDO
- (Mirror comes online on Standard Edition)
- Estimating failover time
 - <http://msdn.microsoft.com/en-us/library/ms187465.aspx>

Adding Files on the Principal

- When adding files to the principal database, the file path must also exist on the mirror
- If it does not, the replay of the log record will fail and the mirroring session will be suspended
- Fix:
 - Remove mirroring
 - Backup the filegroup on the principal with the new file
 - Backup the log so that the log backup contains the ALTER DATABASE command
 - Restore the filegroup backup on the mirror WITH NORECOVERY and WITH MOVE
 - Restore the log backup on the mirror WITH NORECOVERY and WITH MOVE (yes, restoring log WITH MOVE)
 - Restart mirroring

Demo

Mirroring failures and failovers

Losing the Mirror



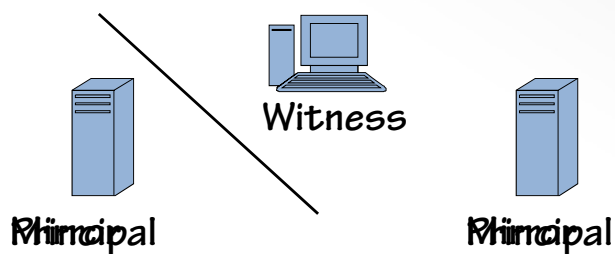
- **Losing the Mirror (NOT Failover)**
 - User connections continue on Principal
 - Principal has quorum with Witness
 - Mirror considered in SUSPENDED state
- **If the Mirror comes back online**
 - Attempt to synchronize (synchronizing) and once synchronized back in HA state

Losing the Principal



- **Losing the Principal (failover)**
 - Mirror has quorum with Witness
 - User connections lost on Principal
 - Mirror brought online as Principal and Principal is considered to be Mirror in SUSPENDED state
- **If the Principal comes back online**
 - Attempt to synchronize and once synchronized back in HA state, as Mirror

Principal Loses Connectivity



- **Losing connectivity with the Principal**
 - Mirror has quorum with Witness
 - Principal has lost connectivity with Witness and Mirror yet is still up and running
 - Principal automatically shuts-down to avoid split brain
 - Mirror brought online as Principal
- **If the Principal comes back online**
 - Attempt to synchronize and once synchronized back in HA state, as Mirror

Mirror Loses Connectivity



- **Witness and Principal should ALWAYS be able to communicate:**
 - Principal always services client requests
 - If site connectivity occurs witness will hold quorum with Principal until Mirror available
- **If connectivity is good then location of Witness is not as critical**

Losing the Witness



- **Losing the Witness**
 - User connections continue on Principal
 - Mirror stays synchronized
- **If the Witness comes back online then back in High Availability**
- **With the Witness offline, additional secondary failures run without automatic failover. If the Mirror becomes unavailable, the Principal stops (to avoid split-brain)!**

High Protection: Losing the Mirror



- No Witness defined (Witness = NULL) = High Protection
- Losing the Mirror
 - Principal CONTINUES to handle user requests
 - No possibility of split brain

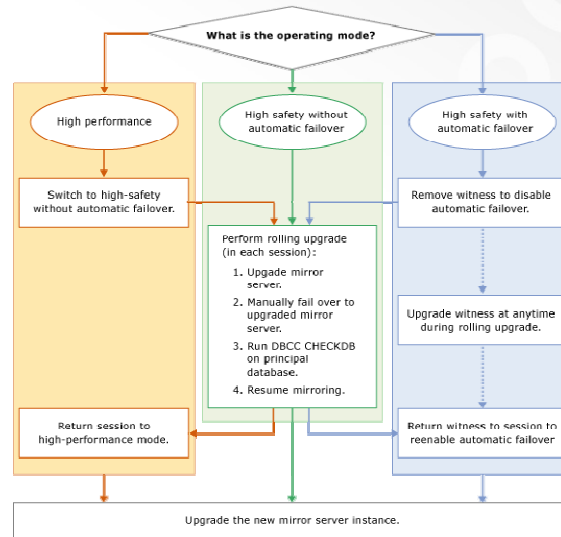
High Protection: Losing the Principal



- No Witness defined (Witness = NULL) = High Protection
- Losing the Principal
 - Mirror does NOT automatically handle user requests
 - Can MANUALLY failover to Mirror by recovering the database, with no data loss if up-to-date
 - Make sure old Principal can see new Principal if it comes online, otherwise it will remain a Principal too -> split-brain!

Rolling Upgrade

- From Books Online:
applies to any
upgrade requiring
SQL to be offline



Mirroring vs. Clustering

- Recovery time
 - Database mirroring: almost immediate failover (typically, only seconds), but how?
 - Real-time rolling-forward of database changes from the principal server to standby server
 - The mirrored server is fully cached and ready to accept workloads because of its synchronized state
 - Failover clustering: requires restart recovery
- Database vs. instance protection
- Regardless of technology, session and transaction state (pending at the time of the failure) is lost and must be reestablished by the application

Mirroring vs. Log Shipping

- **Hardware**
 - Both servers have a complete copy
 - No special hardware requirements
 - Network speed and disk I/O throughput
- **Recovery time**
 - Comparable in TIME only if just brought online
 - NOT comparable in work loss exposure!
- **Work loss exposure**
 - Synchronous mirroring: NONE
 - Asynchronous mirroring: SOME
 - Log shipping: last log available... SOME +, actual amount depends on latency!

Database Snapshot on a Mirror

- A mirror database is unusable directly
- A checkpoint CANNOT be done on a mirror yet database snapshots are allowed
- When you create a database snapshot on a mirror, no checkpoint is performed (it would normally be done on the principal)
- In order to guarantee consistency, the snapshot has to be evaluated back to the most recent checkpoint (that was recorded on the principal) and then perform REDO + UNDO to the LSN of the snapshot

Database Snapshot on a Mirror

- **I/O performance consideration**
 - Make sure database snapshot is not contending for I/O on the mirror's log
- **Failover and snapshots**
 - Manage I/O by making new database snapshot on the new mirror when it comes back online
 - Or failover back to the original principal

Transparent Client Redirect

- **.Net (2.0+) Data Provider for SQL Server supports mirroring**
 - Implicit mode:
 - Developer takes no action
 - Client receives mirror name from principal and caches it
 - .Net will automatically try the mirror if the principal is available
 - If the client reboots after failover, cached info is lost and connections will fail
 - Explicit mode:
 - More robust than implicit mode
 - Developer supplies the name of the failover partner server in the `ConnectionString` property of a `SqlConnection` object
 - Use the following syntax where `<server name>` is the name of the partner server:
- **"`;Failover Partner=<server name>`"**

Performance Implications

- Latency on the principal when using synchronous mode depends on a few things:
 - The mirror's ability for I/O on the mirrored database's transaction log
 - The network time required for the round trip
- Mirror should be comparable to principal – especially critical in failover

Network Latency

What is the network latency between your mirroring principal server and mirror server? And what kind of database mirroring are you using?

Synchronous DBM, 0-50ms latency		31%	14
Synchronous DBM, 50-100ms latency		4%	2
Synchronous DBM, 100-500ms latency		4%	2
Synchronous DBM, > 500ms latency		4%	2
Asynchronous DBM, 0-50ms latency		20%	9
Asynchronous DBM, 50-100ms latency		9%	4
Asynchronous DBM, 100-500ms latency		0%	0
Asynchronous DBM, > 500ms latency		4%	2
Using DBM but no idea of latency		13%	6
Latency is too high to use either kind of DBM		9%	4
Total: 45 responses			

- Source: <http://www.sqlskills.com/BLOGS/PAUL/post/Importance-of-network-latency-when-using-database-mirroring.aspx> (<http://bit.ly/fyEHsf>)

Monitoring

- Database Mirroring Monitor
- **sp_dbmmonitor* procs**
 - Monitoring job independent of Database Mirroring Monitor
- **Mirroring DMVs**
 - sys.database_mirroring
 - sys.database_mirroring_endpoints
 - sys.database_mirroring_witnesses
 - sys.dm_db_mirroring_connections
- Perfmon counters
- WMI event
- Event Notifications

Key Performance Counters

2005+

- Bytes Sent/Received/Sec
- Pages Sent/Sec
- Sends/Receives/Sec
- Log Bytes Sent/Received/Sec
- **Log Send Queue**
 - Extremely important: should be as low as possible
- Redo bytes/Sec
- **Redo Queue**
 - Extremely important: should be as low as possible
- **Transaction Delay**
 - Extremely important: should be as low as possible

2008+

- Log Compressed Bytes Sent/sec - Rcvd/sec
 - Measure compression ratio
- **Log Harden Time (ms)**
 - Measures delay on mirror for a portion of log stream to be hardened on disk
- **Mirrored Write Transactions/sec**
 - Counts transactions that had to wait for a commit record to harden on the mirror
- **Send/Receive Ack Time**
 - Round-trip time between principal and mirror
- **Log Remaining/Scanned for Undo KB**
 - Tracks the UNDO worked needed during a failover

Demo

Monitoring database mirroring

Automatic Page Repair

- Enterprise and Standard Edition 2008+
- SQL Server 2008 ties in the detection of a corrupt page to database mirroring so that corrupt pages can possibly be automatically repaired
- Works for 824 'soft-I/O' errors, some 823 'hard-I/O' errors
 - Errors are hit by queries reading the page
- Corrupt pages on the principal AND mirror can be repaired
- Repairs are asynchronous, corrupt pages are unusable until fixed
- NOTE:
 - This is only meant to be a band-aid to prevent downtime
 - It is NOT a substitute for having alerts for high severity errors and taking affirmative action to rectify/prevent them

Automatic Page Repair

- **Mechanism**

- Corrupt page is hit and logged in msdb.dbo.suspect_pages
- Page is requested from the other mirroring partner, sent over if available, and repaired with the new image
- The page is marked as fixed in msdb.dbo.suspect_pages

- **Additional information:**

- DMV sys.dm_db_mirroring_auto_page_repair allows tracking of previous 100 corrupt pages found in all mirroring sessions
- Error log output traces what the mirroring partnership is doing
 - 2011-09-27 17:23:19.96 spid26s Database mirroring is attempting to repair physical page (4:4256) in database "TicketSalesDB" by requesting a copy from the partner.
 - 2011-09-27 17:23:20.42 spid26s Database mirroring successfully repaired physical page (4:4256) in database "TicketSalesDB" by obtaining a copy from the partner.
- SQL Server Profiler "Database Suspect Data Page" Event Class



45

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

Demo

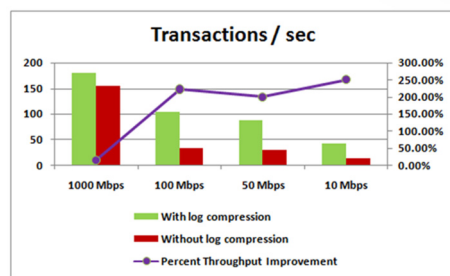
Automatic page repair

Log Compression

- Enabled by default on Enterprise and Standard Edition 2008+
- Compressing the log before sending it, send more log over the same bandwidth BUT at the expense of CPU
 - CPU is needed to compress AND decompress the log
- For high-bandwidth networks, could allow previously unpalatable maintenance operations to be performed, that are always fully logged with mirroring (e.g. index rebuild)
- It can be switched off if required, using trace flag 1462
- NOTE: compression ratio (and hence potential throughput gain) depends on the data being processed by the application
- Hardware compression by network may be better, but may be compromised by leaving log stream compression enabled

Effect of Log Compression

- Comparison of transactions per second against network bandwidth, with and without log compression

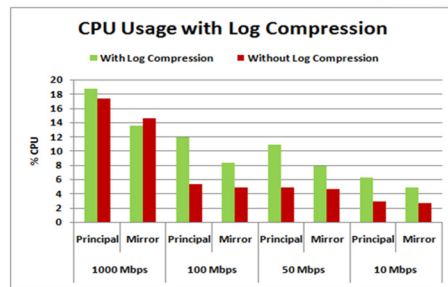


- Percentage increases in throughput is most dramatic for low network bandwidths

* Source: Database Mirroring Log Compression in SQL Server 2008 Improves Throughput

Cost of Log Compression

- Comparison of CPU usage against network bandwidth with and without log stream compression



- CPU goes up when compression is on, both because of compression/decompression, but also because the server can now process more transactions per second

* Source: Database Mirroring Log Compression in SQL Server 2008 Improves

Database Mirroring and Clustering

- Common between two geographically dispersed clusters
 - Cheaper alternative to SAN replication
- Make sure:
 - The database mirroring partner timeout is set to be higher than the usual cluster failover time to avoid unwanted mirroring failovers
 - Note: you may get mirroring failovers if you manually failover SQL Server
 - As the Windows Server is still available to respond to the mirror's ping and say nothing is listening on the mirroring port
 - Remove witness before manual cluster failover to prevent spurious mirroring failovers

Database Mirroring and Log Shipping

- **Common to have one or more log shipped secondaries as warm standbys**
 - Possibly even one with a load delay
- **Must make sure to configure log backup jobs on the principal AND the mirror**
 - The one on the mirror will continually do nothing until a failover occurs
- **More details in the whitepaper Database Mirroring and Log Shipping Working Together**
 - <http://bit.ly/bXsbz6>

Database Mirroring and Replication

- **Publication database mirroring is possible in 2005+**
- **Distribution database mirroring is not possible**
 - Requires clustering to add HA
- **Subscription database mirroring is possible in 2008**
 - Technically possible in 2005, but requires a full reinitialize of the subscription on failover
- **Usually only for transactional replication, not peer-to-peer, as each peer node is a publisher AND subscriber**
- **Whitepaper: SQL Server Replication: Providing High Availability using Database Mirroring**
 - <http://bit.ly/n4PTL>

Review

- Mirroring configuration
- Failure types and speed
- Failure scenarios
- Client redirection
- Monitoring
- Troubleshooting

Database Mirroring Resources

- **Book: Pro SQL Server 2008 Mirroring**
- **Whitepapers**
 - Database Mirroring in SQL Server 2005
 - <http://www.microsoft.com/technet/prodtechnol/sql/2005/dbmirror.mspix>
(<http://bit.ly/PI370n>)
 - Database Mirroring: Alerting on Database Mirroring Events
 - <http://www.microsoft.com/technet/prodtechnol/sql/2005/mirroringevents.mspix>
(<http://bit.ly/QCv6FU>)
 - Database Mirroring: Best Practices and Performance Considerations
 - http://www.microsoft.com/technet/prodtechnol/sql/2005/technologies/dbm_best_pract.mspix (<http://bit.ly/PI3bxd>)
 - Implementing Application Failover With Database Mirroring
 - <http://www.microsoft.com/technet/prodtechnol/sql/bestpractice/implappfailover.mspix> (<http://bit.ly/QCvdKO>)

Questions?

