Microsoft SQL Server 2008

High Availability and Disaster Recovery with SQL Database Mirroring

Rolf Tesmer
Principal SQL Consultant

Ben Thurecht
Senior SQL Consultant

Agenda

- SQL Server HA / DR Technologies Overview
- Database mirroring terminology, constraints, operating modes, internal operation, monitoring
- Database Mirroring Live Demonstrations

1. Setup of Basic Database Mirroring
2. Synchronous vs. Asynchronous Modes
3. Using the Automatic Client Redirect
4. Automatic Recovery from Page Corruption

High Availability Technologies in SQL 2008

- There are several technologies in SQL Server 2008 which allow us to build Highly Available SQL database platforms:
  - Microsoft Clustering (and its 3rd Party variations) (HA)
  - Microsoft SQL Log Shipping (DR)
  - Microsoft SQL Replication (HA / DR)
  - SAN based replication technologies (DR)
  - Virtualisation Technology (VMware, Hyper-V) (HA / DR)
  - Microsoft SQL Database Mirroring (HA / DR)

When to Use Which Availability Solution?

- ***As usual it depends...***
- **Selection is driven by 2 key metrics set by the business**

RPO = Recovery Point Objective	RTO = Recovery Time Objective
How much data can you afford to lose?	*How long can you afford to be down?*

- **Other questions that can drive the solution**
  - **Do you need cover for a database, instance, server or site**
  - **Do you need just local availability or off site redundancy**
  - **Is a DR site available, what is the cross-site link capacity**
  - **Is a SAN available, if so at what sites, what is the capacity**
  - **Is any virtualisation technology available**
  - **How big is your budget!**

How Do These Technologies Compare?

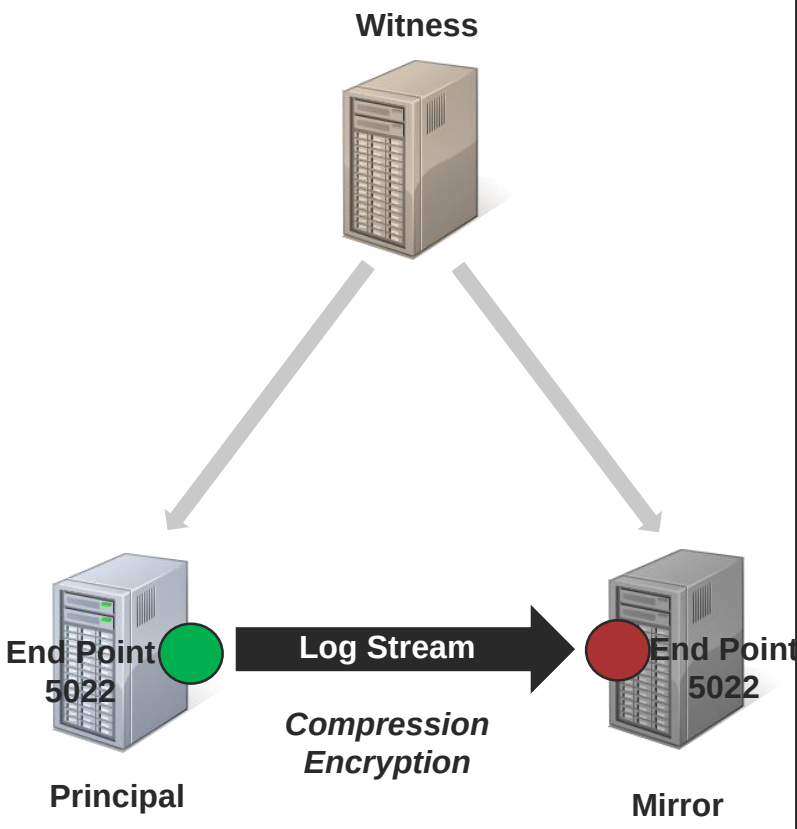
Factor	Clustering	Log Shipping	Transaction Replication	SAN Mirroring	SQL DB Mirroring
Failover Time (RTO)	Fast	Medium	Fast	Medium	Fast
Potential Loss (RPO)	Near Zero uncommitted	Up to 1 min	Up to 1 min	Near Zero uncommitted	Near Zero uncommitted
Prod Overhead	N/A (HA Only)	< 0 - 2 %	< 0 - 5%	< 0 - 2%	< 0 - 2%
Prod to DR Bandwidth	N/A (HA Only)	Minimal change only	Minimal change only	Minimal change only	Minimal change only
Single Point of Failure	Shared Disk	None	None	None	None
Day-to-Day DBA Time	Near Zero	Medium	High	Near Zero	Minimal

Database Mirroring Overview

- So what is SQL Database Mirroring?
  - A technology introduced in SQL 2005 (SP1) which allows a database to be "synchronised" to another SQL 2005/8 instance
  - Similar concept to log shipping, but easier to configure and manage
  - It is a powerful tool that can be used for:
    1. High Availability (HA)
    2. Disaster Recovery (DR)
    3. Reporting (via usage of Database Snapshots at Mirror site)
    4. Rolling upgrades (SQL 2005/2008 patching, SQL 2005 to 2008)
  - Available in SQL Server Standard and Enterprise Editions

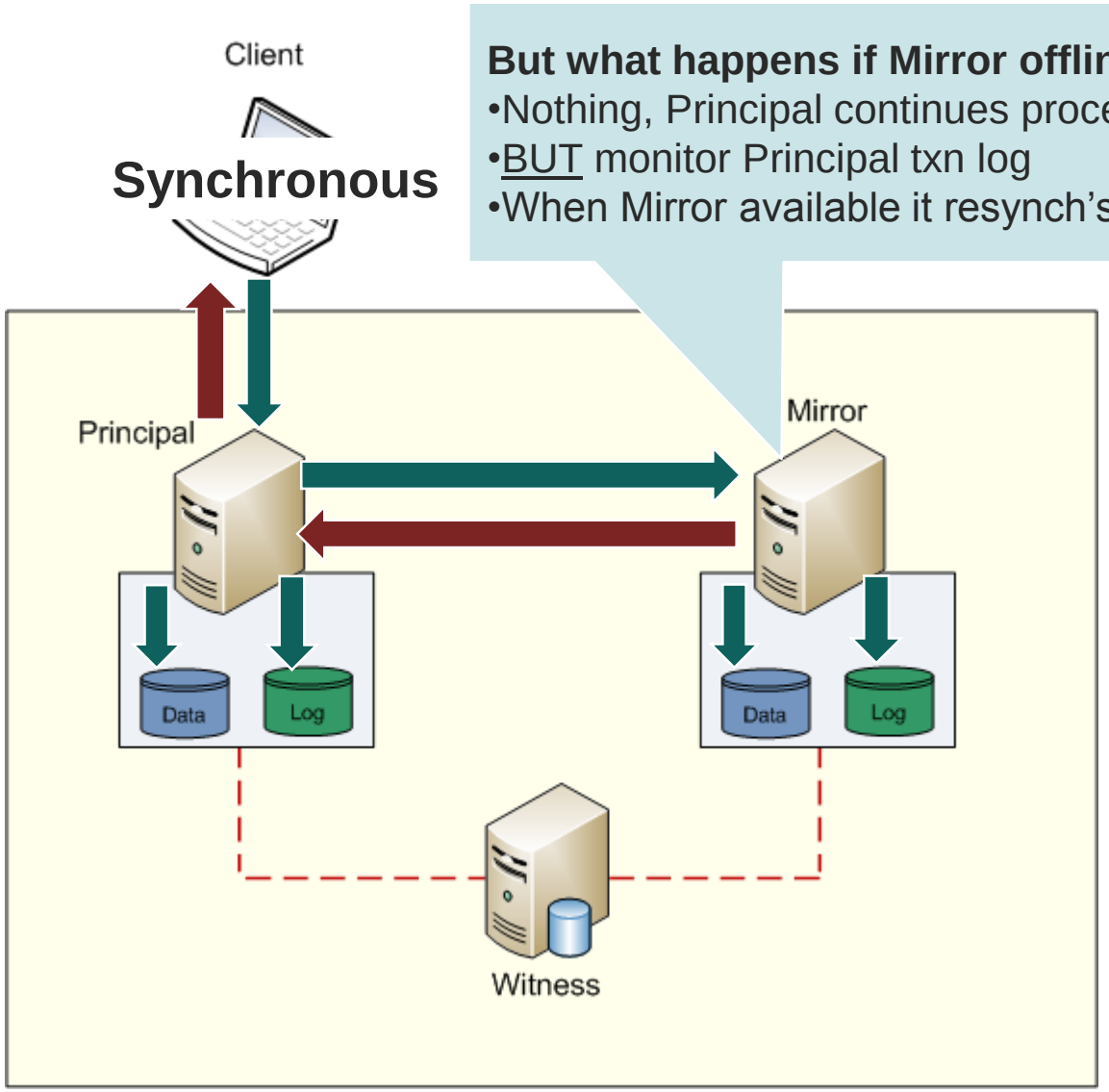
Some Basic Mirroring Terminology

- The source database / instance is know as the Principal
- The destination database / instance is known as the Mirror
- The third (optional) component is the Witness
- A Mirroring Endpoint is a specially configured TCP port (5022) on the SQL Servers which allows a server to participate in Database Mirroring
- Log Stream is the flow of data change from Principal to Mirror. SQL 2008 compresses and encrypts (AES/RC4) this data by default.



General Database Mirroring Process Overview

1. **Client Submits an update Transaction**
2. **Principal Commits to Log and sends transaction to mirror**
3. **Mirror commits to log**
4. **Mirror sends acknowledgement**
5. **Principal returns control to client**
6. **Data is flushed from buffer cache to disk by Lazy Writer**



But what happens if Mirror offline?
- Nothing, Principal continues processing
- BUT monitor Principal txn log
- When Mirror available it resynch's data

Basic Information on Database Mirroring

1) Operates at the Database level only, does not replicate any other SQL component
2) The principal database must be in “Full Recovery” mode, any DB Compatibility setting is OK
3) Cannot mirror databases with FILESTREAM, however FT, TDE, replication, clustering is OK
4) There can only be one mirrored copy per principal database
5) The principal / mirror databases must have the same name, therefore different SQL instances
6) The principal / mirror instances need to use Windows Auth (AD) or Certificate Auth (non-AD)
7) The principal / mirror must be the same Edition (recommend latest SP where possible)
8) Standard Edition is Synchronous only, Enterprise is Synchronous or Asynchronous
9) The mirror database is in “Recovering” state so it cannot be directly accessed (but can take SNAPSHOT of mirror copy for read only purposes)
10) Is dependent on a reliable network link and server infrastructure (both SQL instances)
11) Log Stream Compression on by Default, gives ~20% TPS boost. Can disable with –T1462.
12) Theoretical limit of 10 DB’s on x86, ~ unlimited (to max DB’s) on x64
13) Beware of mass data change at the Principal (reindex, bulk load, etc)

Database Mirroring Operating Modes

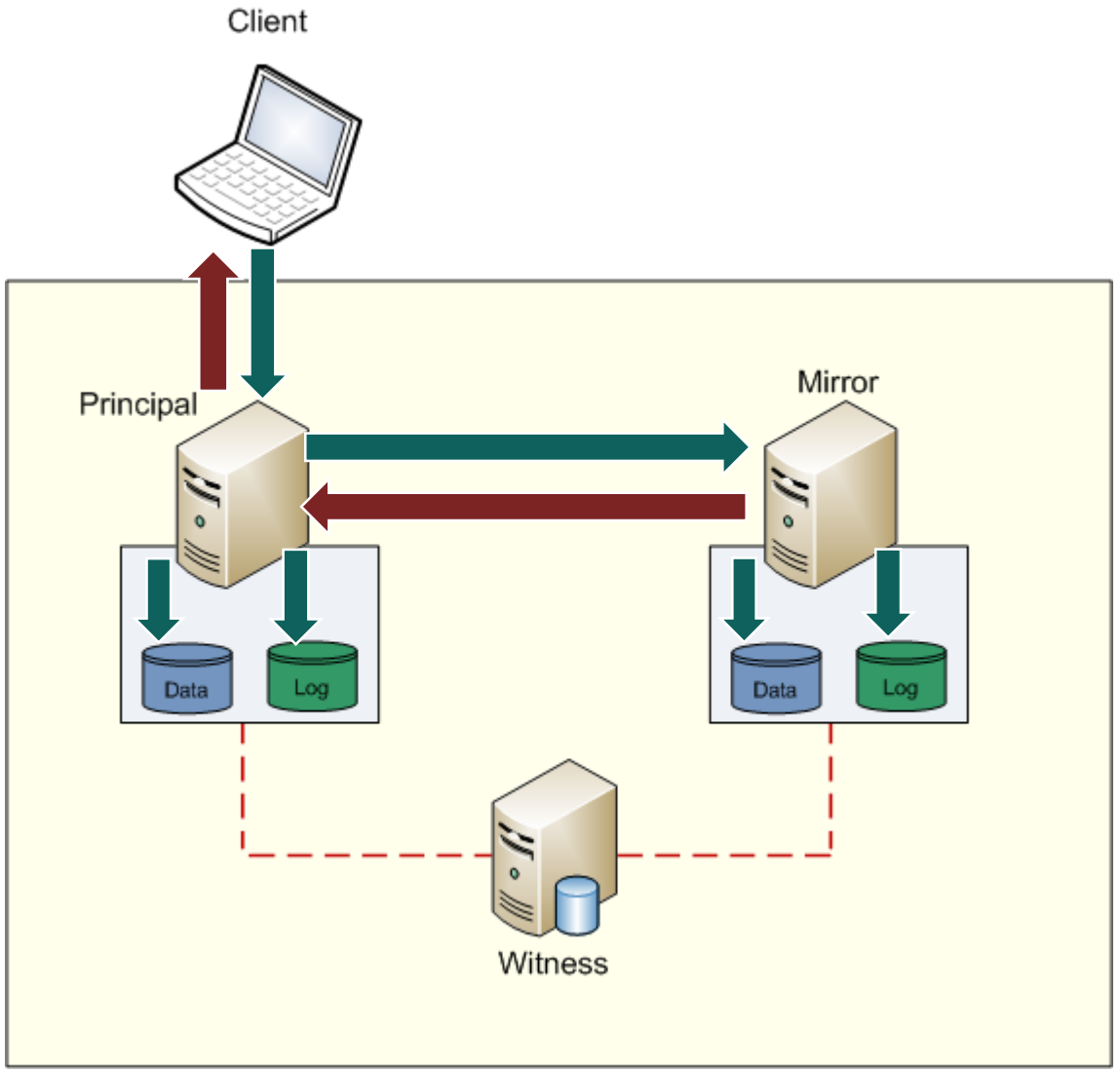
1. **High Performance**
   - ➔ Enables the best performance for Database Mirroring
   - ➔ Uses Asynchronous data transfer (Enterprise Only)
2. **High Protection**
   - ➔ Enables the best recovery option for Database Mirroring
   - ➔ Uses Synchronous data transfer (Enterprise and Standard)
3. **High Availability**
   - ➔ Enables the best availability for Database Mirroring
   - ➔ Uses Synchronous data transfer (Enterprise and Standard)

Operating Mode: High Performance

- Uses asynchronous data transfer
- Does not wait for transaction commit at Mirror
- Fastest option, no client performance degradation
- BUT data loss is possible
- Failover is a manual process (i.e. a business decision is made)
- Only available in SQL Enterprise Edition

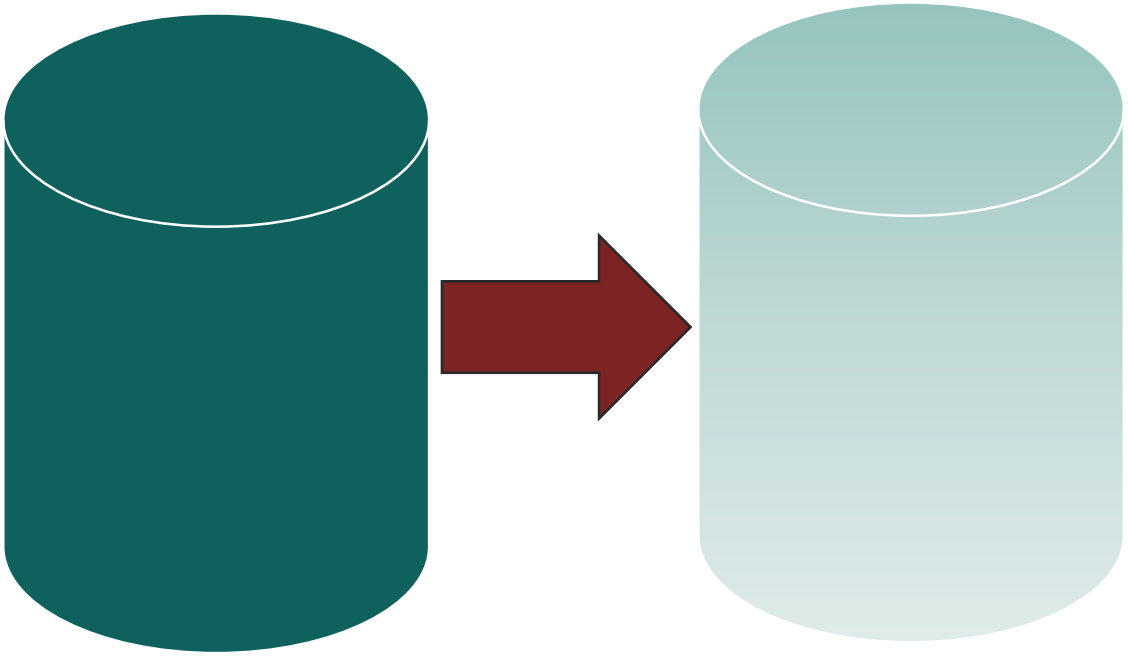
Asynchronous Data Transfer Overview

1. Client Submits an update Transaction
2. Principal Commits to Log and returns control to the client
3. Transaction is sent to Mirror
4. Mirror commits to log
5. Mirror sends acknowledgement
6. Data is flushed from buffer cache to disk by Lazy Writer



Demonstration #01

- Implementing High Performance (Asynchronous) Database Mirroring

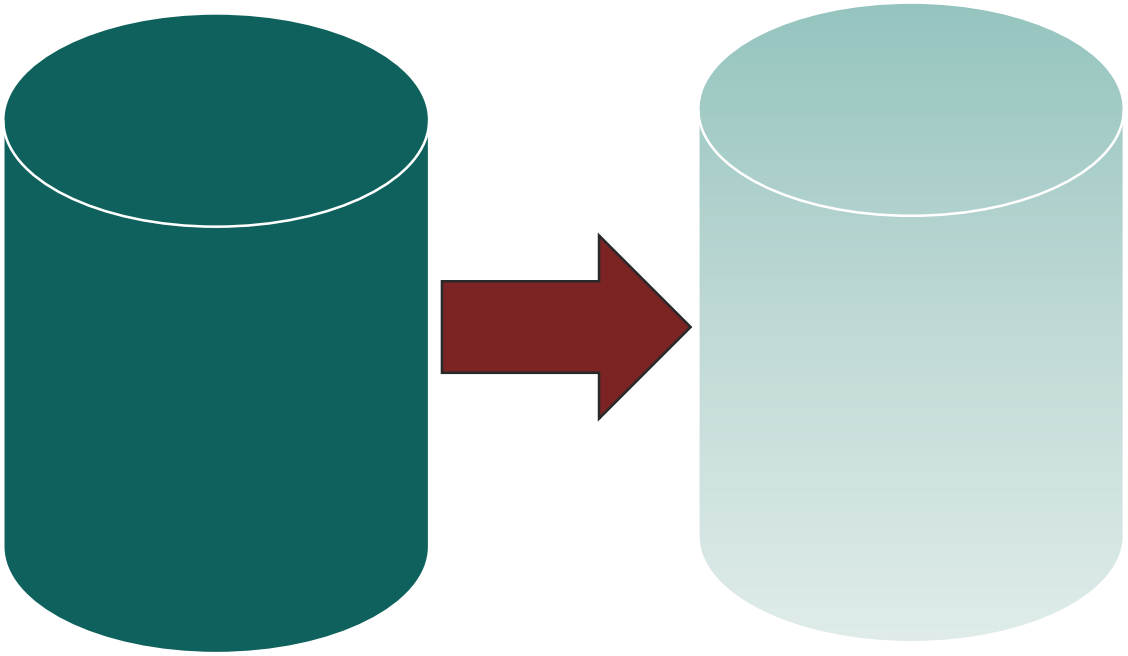


Operating Mode: High Protection

- Uses synchronous data transfer
- Waits for transaction commit at Mirror
- Can lead to Performance degradation and this is directly tied to network latency and bandwidth
- Data loss is not possible (assuming Mirror is available)
- Failover is a manual process (i.e. a business decision is made)

Demonstration #02

- Synchronous vs. Asynchronous Data Transfer

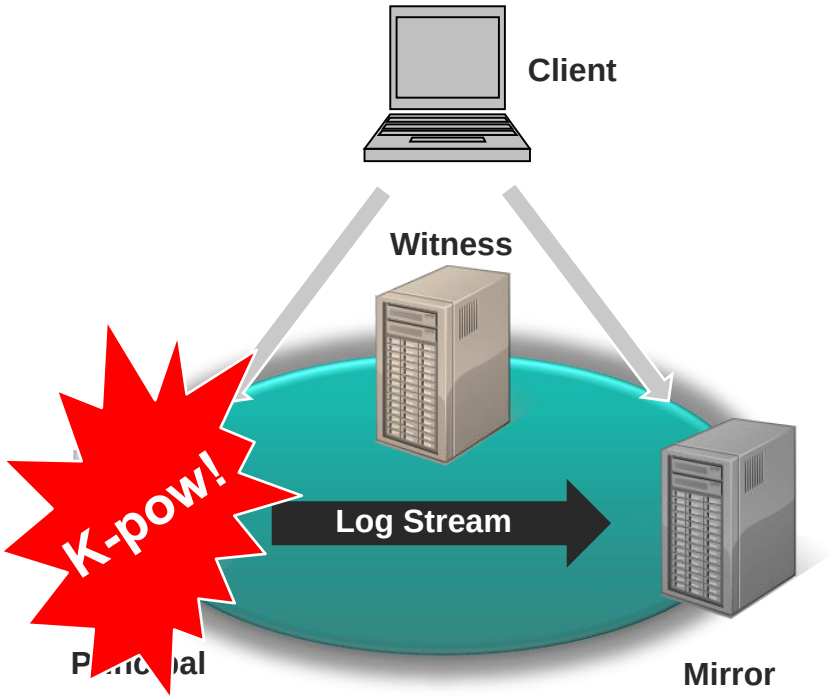


Operating Mode: High Availability

- Uses synchronous data transfer
- Waits for transaction commit at Mirror
- Performance degradation directly related to network latency and bandwidth
- Data loss is not possible (assuming Mirror is always available)
- Failover is automatic via a 3rd Witness server (i.e. SQL Server makes the decision for the business)

What is Automatic Client Redirect

- Automatic Client Redirect is a client side technology which allows an application to be "Database Mirroring Aware"
- Allows application to detect a mirroring failover event and automatically reconnects to the new Principal server



- The client could be anything from a .NET Website to a thick-client desktop application
- *...Add* High Availability Database Mirroring Mode (i.e. using a witness server), a highly available n-tier application can be built

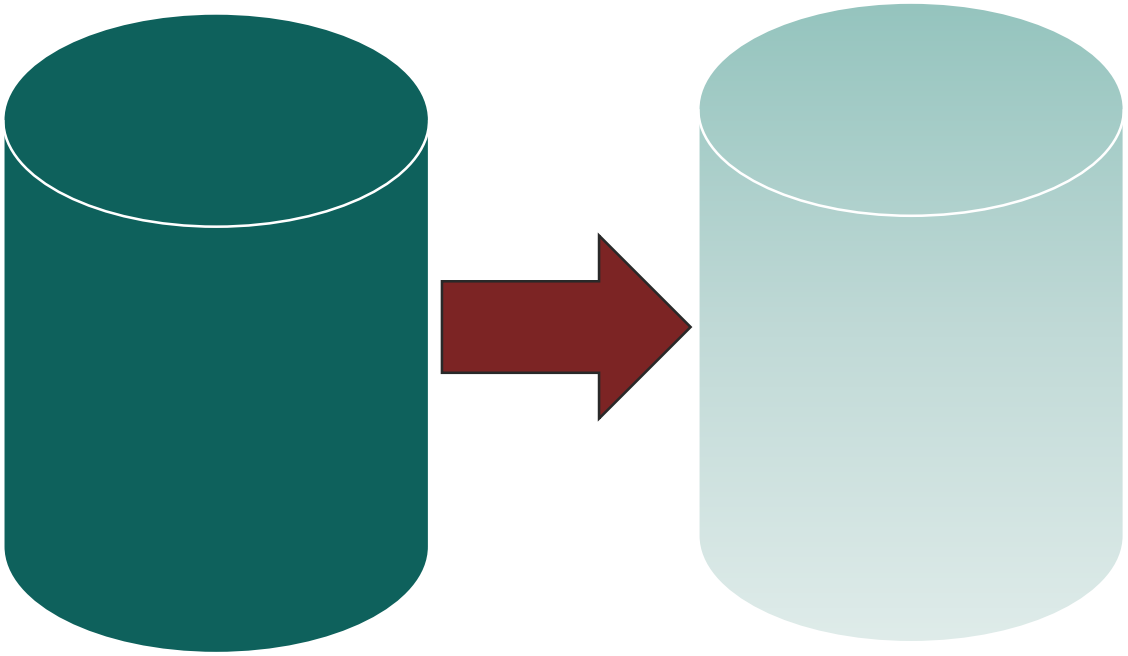
Using Automatic Client Redirect

- Need to use new SQL connectivity libraries, namely the SQL Native Client (SNAC) or the .NET data provider for SQL Server
- Legacy applications can't take advantage of this feature (i.e. ADO)
- Example Application Connection String:

```
"Server = MyServer_P, 1433;
Failover Partner = MyServer_M, 1433;
Database = MyDatabase;
Integrated Security = SSPI";
```

Demonstration #03

- Automatic Client Redirect



“ALTER DATABASE” T-SQL Command

```
ALTER DATABASE database_name

  SET { <partner_option> | <witness_option> }

  <partner_option> ::=
      PARTNER { = 'partner_server'
      | FAILOVER
      | FORCE_SERVICE_ALLOW_DATA_LOSS
      | OFF
      | RESUME
      | SAFETY { FULL | OFF }
      | SUSPEND
      | TIMEOUT integer
  }

  <witness_option> ::=
      WITNESS { = 'witness_server'
      | OFF
  }
```

- Only available on Mirror
- Forces SQL to use Mirror
- Principal must be offline
- Suspends mirror session
- Can lose log stream data

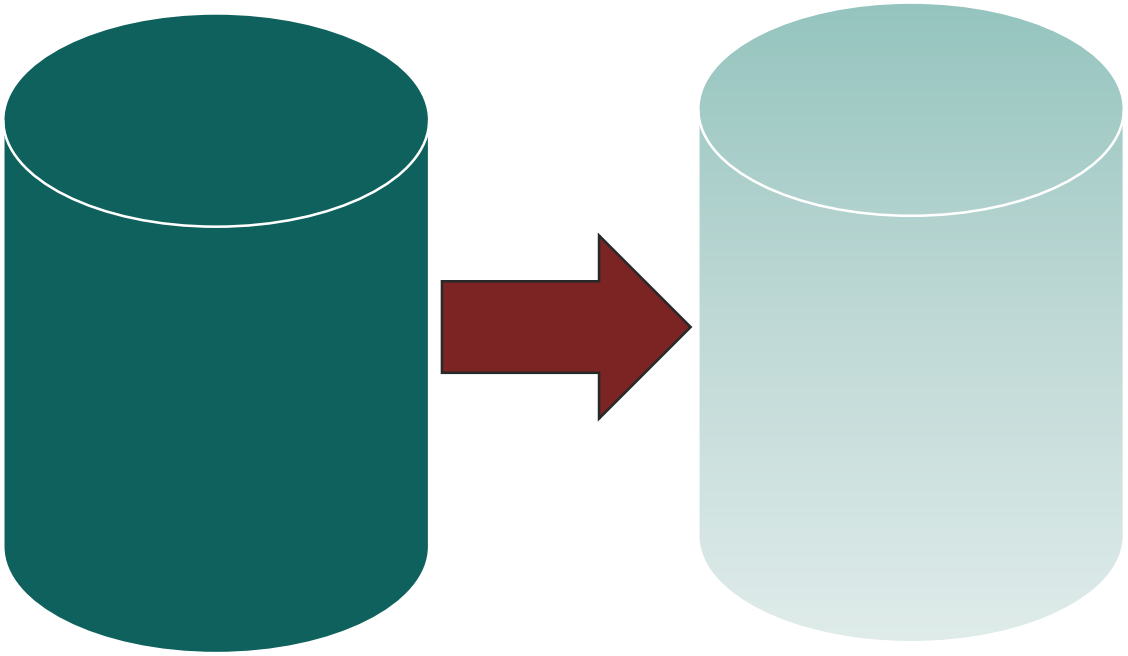
- Specifies “PING” period
- Default is 10 sec
- Max time Mirror will wait
- Keep above 5 sec!

Automatic Page Repair (SQL 2008 only)

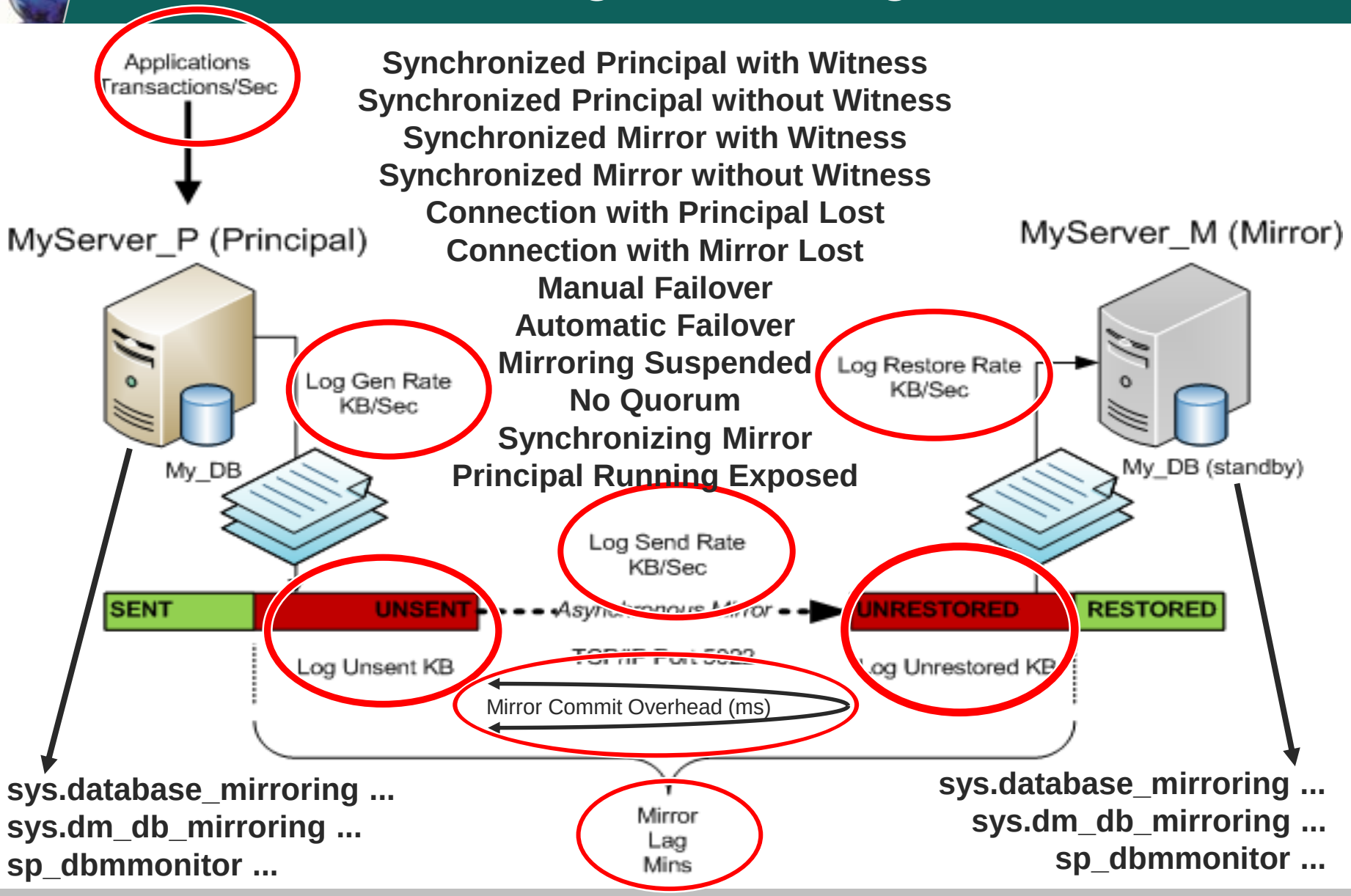
- New feature in SQL 2008 is automatic page repair
- SQL Server automatically detects corrupt pages on the Principal when data is “touched” (i.e. Selected) and transparently recovers them from the Mirror
- Does this without DBA intervention but logs message in SQL Error log
  - `Database mirroring is attempting to repair physical page (<FileID>:<PageID>) in database "<DatabaseName>" by requesting a copy from the partner.`
  - `Database mirroring successfully repaired physical page (<FileID>:<PageID>) in database "<DatabaseName>" by obtaining a copy from the partner`
- DOES NOT replace the need for DBCC CHECKDB

Demonstration #04

- Automatic Page Repair

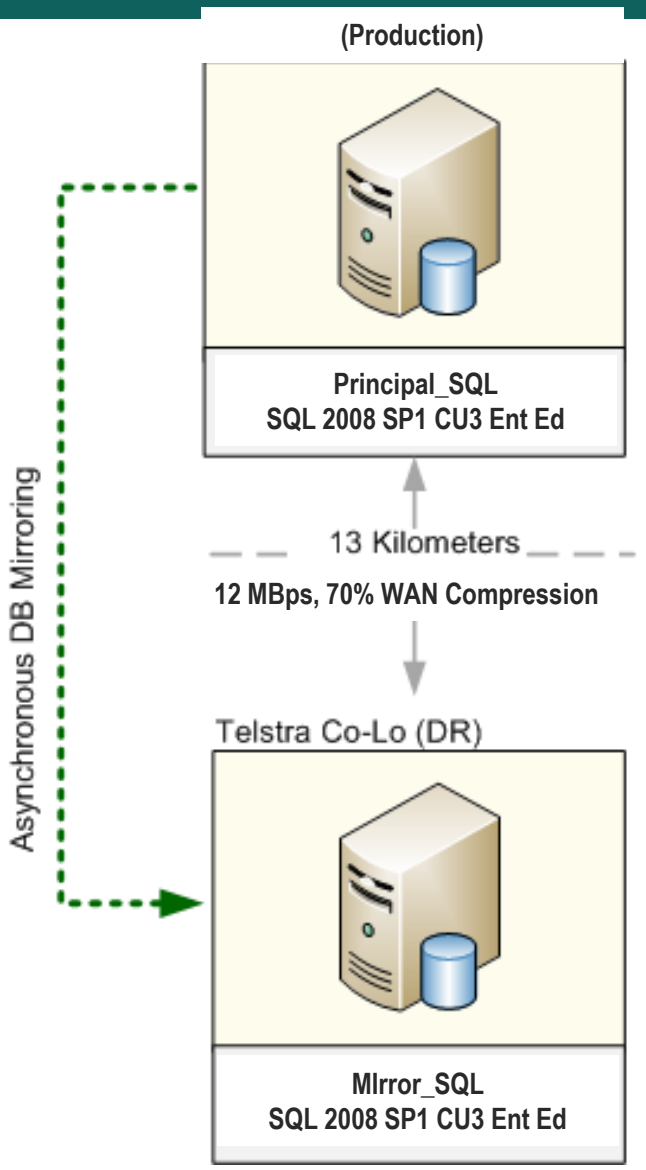


Database Mirroring Monitoring



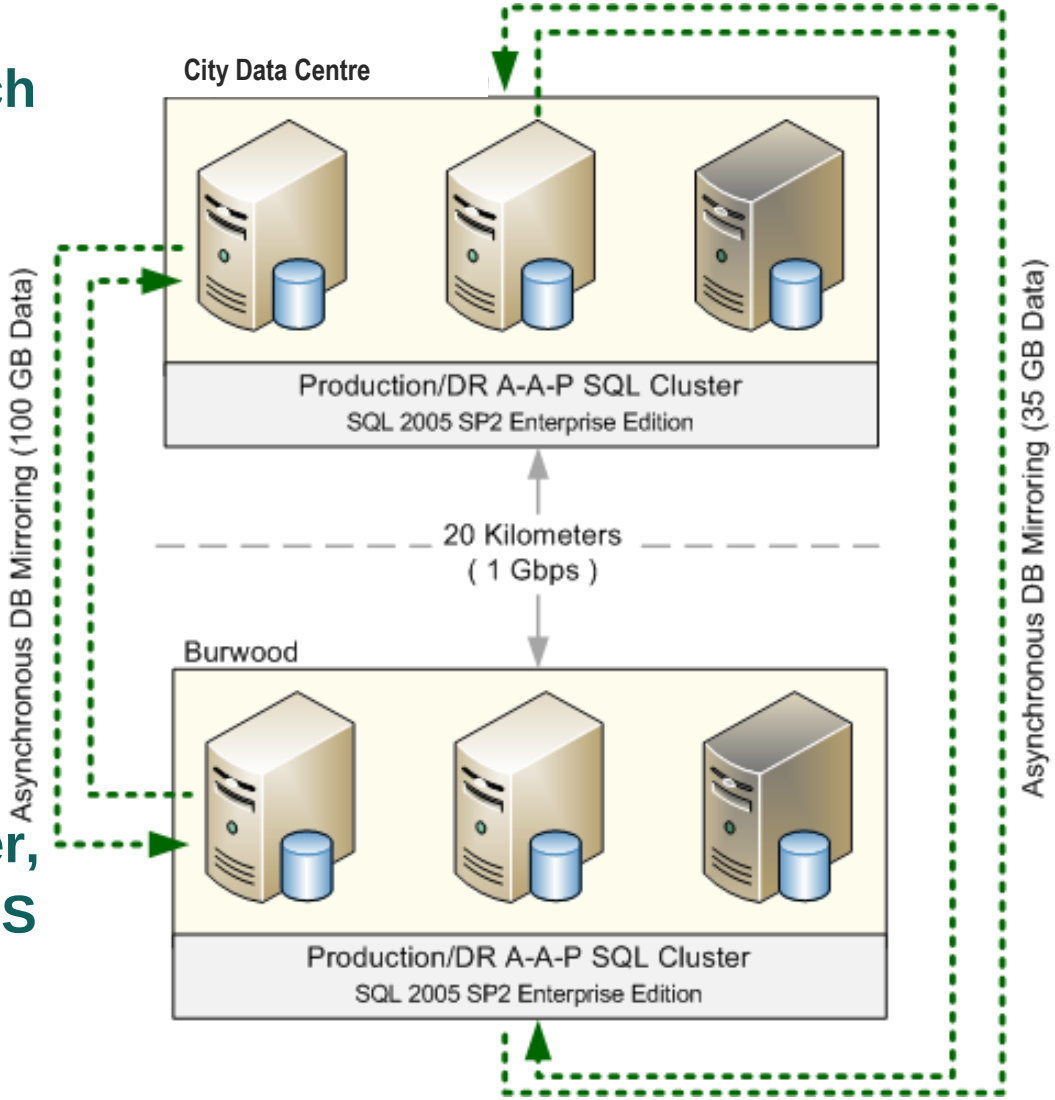
Case Study: CLIENT-A

- All Transactions from CLIENT-A run through Asynch Database Mirroring session
- 3000+ Users
- 1200 batches / second
- 1.6TB database
- 50GB+ daily data change (mostly out of core business hours)
- Riverbed WAN optimisation appliances at each end (>90%)
- Only used for off site data protection / redundancy



Case Study: CLIENT-B

- Mix of Asynch and Synch database mirroring
- Supports critical management systems
- Supports SQL Server Reporting Services (SSRS)
- Supports MOSS 2007
- Full Recovery on failover, client redirection via DNS changes



Database Mirroring Summary

High Performance

- Asynchronous Data Transfer
- Data Loss is possible
- Does not require a Witness Server
- Manual Failover
- No performance degradation
- Enterprise Edition Only

High Availability

- Synchronous Data Transfer
- Data Loss not possible
- Requires a Witness Server
- Automatic Failover
- Possible performance degradation
- Standard and Enterprise Edition
- Can be combined with automatic client redirect to build highly available applications

High Protection

- Synchronous Data Transfer
- Data Loss not possible
- Does not require a Witness Server
- Manual Failover
- Possible performance degradation
- Standard and Enterprise Edition

What Mirroring Configuration is Best for me?

- ***As usual it depends…***
- **On the business Recovery Point Objective (RPO)**
  - How much data can the business afford to lose?
- **On the business Recovery Time Objective (RTO)**
  - How long can the business afford the database to be down?
- **On the business budget**
  - Can you afford to utilise SQL 2008 Enterprise Edition?
- **On the technical infrastructure**
  - What server, network and storage infrastructure is available?
- **All these considerations determine the suitable mirroring solution**
- **And of course test, test, test… and then TEST some more!**

Technical References

Database Mirroring in SQL Server 2005

http://www.microsoft.com/technet/prodtechnol/sql/2005/dbmirror.mspx

SQL Server 2005 Database Mirroring Best Practices and Performance Considerations

http://technet.microsoft.com/en-us/library/cc917681.aspx

Implementing Application Failover with Database Mirroring

http://www.microsoft.com/technet/prodtechnol/sql/bestpractice/implappfailover.mspx

SQL Server 2008: Automatic Page Repair with Database Mirroring

http://www.sqlskills.com/BLOGS/PAUL/post/SQL-Server-2008-Automatic-Page-Repair-with-Database-Mirroring.aspx

SQL Server 2008 Log Stream Compression Performance

http://www.sqlskills.com/blogs/paul/post/SQL-Server-2008-Performance-boost-for-Database-Mirroring.aspx

Things to consider when setting up database mirroring in SQL Server

http://support.microsoft.com/default.aspx/kb/2001270

SQL Server 2005 and 2008 Books Online

Microsoft