**SQL Server High Availability:**

**Quick view of Database Mirroring session between partners**

From SQL Server 2005 version onwards database mirroring has been a good resource to provide High Availablity for the systems without having special hardware requirement.

While I was assisting at the SME lounge within Database Platform area in Tech-Ed North America conference, I had few interesting questions from the users asking about various methods to monitor database mirroring environment.

Database Mirroring Monitor is the tool which is a first hand resource to see what is happening within the partners, also the other methods such as query against DBM metada in catalog views, use of dynamic management views, PERFMON counters for DBM, DBM related event notifications and server events using WMI provider. To go further about catalog views and DMVs to view the mirroring information you can get information for each mirrored database from the instance. Here is a little TSQL to obtain commonly used mirroring metadata:

SELECT d.name, d.database\_id, m.mirroring\_role\_desc,

       m.mirroring\_state\_desc, m.mirroring\_safety\_level\_desc,

       m.mirroring\_partner\_name, m.mirroring\_partner\_instance,

       m.mirroring\_witness\_name, m.mirroring\_witness\_state\_desc

FROM   sys.database\_mirroring m JOIN sys.databases d

ON     m.database\_id = d.database\_id

WHERE  mirroring\_state\_desc IS NOT NULL

**The** above query uses the sys.database\_mirroring & sys.database catalog views, which can be used on principal side or mirror side instance to obtain the information and further to get witness related information you an use sys.database\_mirroring\_witness catalog view. A sample TSQL to get list the corresponding principal and mirror server names, database name, and safety level for all the mirroring sessions for which this server is a witness.

SELECT principal\_server\_name, mirror\_server\_name,

       database\_name, safety\_level\_desc

FROM   sys.database\_mirroring\_witnesses

ENDPOINT is the key in database mirroring, as the connection management between SQL Server instances (since 2005 version) is based on *endpoints*. BOL defines it as "*An endpoint is a SQL Server object that enables SQL Server to communicate over the network. For database mirroring, a server instance requires its own, dedicated database mirroring endpoint. All mirroring connections on a server instance use a single database mirroring endpoint. This endpoint is a special-purpose endpoint used exclusively to receive database mirroring connections from other server instances"*.  As the endpoint uses a specific port the usual catalog view of sys.database\_mirroring\_endpoints will not maintain the information, it is in sys.tcp\_endpoints view, so here is another TSQL to join both of these catalog views to obtain information about endpoints:

SELECT e.name, e.protocol\_desc, e.type\_desc, e.role\_desc, e.state\_desc,

       t.port, e.is\_encryption\_enabled, e.encryption\_algorithm\_desc,

       e.connection\_auth\_desc

FROM   sys.database\_mirroring\_endpoints e JOIN sys.tcp\_endpoints t

ON     e.endpoint\_id = t.endpoint\_id

These a just monitoring aspects of a database mirroring pair, but there is much to monitor the performance of database mirroring, SQL Server provides a System Monitor performance object (aka PERFMON) for SQLServer:Database Mirroring related counters on each partner (principal and mirror). The Databases performance object provides some important information as well, such as throughput information (**Transactions/sec** counter). Here is the reference from Technet article about the important counters that are need to watch if you feel that performance has been degraded withint your DBM enviornment.

On the principal:

         **Log Bytes Sent/sec**: Number of bytes of the log sent to the mirror per second.

         **Log Send Queue KB**: Total kilobytes of the log that have not yet been sent to the mirror server.

         **Transaction Delay**: Delay (in milliseconds) in waiting for commit acknowledgement from the mirror. This counters reports the total delay for all the transactions in process at that time. To determine the average delay per transaction, divide this counter by the **Transactions/sec** counter. When running asynchronous mirroring this counter will always be 0.

         **Transactions/sec**: The transaction throughput of the database. This counter is in the **Databases** performance object.

         **Log Bytes Flushed/sec**: The rate at which log records are written to the disk. This is the log generation rate of the application. It plays a very important role in determining database mirroring performance. This counter is in the **Databases** performance object.

         **Disk Write Bytes/sec**: The rate at which the disk is written to. This counter is in the **Logical Disk** performance object and represents. Monitor this counter for the data as well as the log disks.

On the mirror:

         **Redo Bytes/sec**: Number of bytes of the transaction log applied on the mirror database per second.

         **Redo Queue KB**: Total kilobytes of hardened log that remain to be applied to the mirror database to roll it forward.

         **Disk Write Bytes/sec**: The rate at which the disk is written to. This counter is in the **Logical Disk** performance object and represents. Monitor this counter for the data as well as the log disks on the mirror.

Lastly you can take help of event notifications (trace) by using SQL Trace which helps recording specified events, SQL Trace helps you troubleshoot performance, audit database activity, gather sample data for a test environment, debug Transact-SQL statements and stored procedures, and gather data for performance analysis tools by accessing SQLtrace or SQL Profiler. The 2 events are important for DBM related environment such as Database Mirroring state change which indicates the mirroring state of mirrored database changes and Audit Database Mirroring login event class which reports the audit messages related to DBM transport security.