Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 4, Issue. 6, June 2015, pg.387 – 391

RESEARCH ARTICLE

Choosing Between High Availability Solutions in Microsoft SQL Server

Sapna Dahiya, Pooja Ahlawat
M.Tech student, Department of CSE, R.N College of Engineering & Management
Assistant Professor, Department of CSE, R.N College of Engineering & Management
Sapna30july@gmail.com

Abstraction: The purpose of high availability is to restore normal operations as quickly as possible in case of failure or disaster. Disaster is the serious disruption of services either natural or man-made leading to adverse impact on the business. Therefore several high availability techniques have been designed which have their own advantages and limitations. This paper presents a review of some significant work in the area of High Availability. After a brief introduction, some popular approaches are classified into different groups and an overview of various techniques is provided.

1. Introduction

In today's world, all the major businesses are running many critical applications such as financial applications cannot afford fractions of second's downtime. Therefore all the major DBMS players in market like Microsoft, Oracle are focusing in providing reliable high availability solutions. Just imagine what will happen if a bank website will go down. They will incur financial lose in million trillion figures just in few seconds. I am writing down this paper to share information about administering several high availability solutions, which can ensure minimum downtime in case of service failures, or disasters, which Microsoft offers. The latest version of SQL Server, which is available in market, is MS SQL 2012 and biggest enhancement of this version is ALWAYSON, which means no downtime.

SQL Server provides several high availability solutions but choice mostly lies with the business requirements.

2. Several High Availability options include the following

2.1 AlwaysON

AlwaysON is one of the better-known features in the release of SQL Server 2012 Enterprise Edition. It is designed to meet the ever-increasing need for 'High Availability' (HA). AlwaysOn does not use entirely new technologies but makes more effective use of existing technologies that are tried and tested. AlwaysON is designed to keep more control on achieve High Availability.

AlwaysON combine mirroring, logshipping, clustering as a single functionality. In case of crash situations now the production instance can failover to multiple secondaries. Moreover now it is possible to read from secondary. It ensures:

San level Block Replication

Transaction Log Shipping

Database Mirroring

Transactional Replication

Peer-to-Peer Replication

Multi-Site Windows Server Failover Clustering

2.2 Failover Clustering

Up to MS SQL Server 2008R2 Failover clustering provides high-availability support for an entire instance of SQL Server. A failover cluster is a combination of one or more nodes, or servers, with two or more shared disks. Minimum 2 nodes are the prerequisite. All the Applications are installed into a Microsoft Cluster Service (MSCS) cluster group, known as a resource group. Ideally resource group name is given to group all the resources, which SQL cluster is using. It ensures that at the time of failover all the SQL resources failover to the secondary node all-together. And this is not the case that SQL Services are running on one node and agent on another node. Resource group name should be same as Network Name. At any time, each resource group is owned by only one node in the cluster. The application service has a virtual name that is independent of the node names, and is referred to as the failover cluster instance name. An application can connect to the failover cluster instance by referencing the failover cluster instance name. The application does not have to know which node hosts the failover cluster instance.

A SQL Server failover cluster instance appears on the network as a single computer, but has functionality that provides failover from one node to another if the current node becomes unavailable. For example, during a non-disk hardware failure, operating system failure, or planned operating system upgrade, you can configure an instance of SQL Server on one node of a failover cluster to fail over to any other node in the disk group.

A failover cluster does not protect against disk failure. You can use failover clustering to reduce system downtime and provide higher application availability. Failover clustering is supported in SQL Server Enterprise and SQL Server Developer, and, with some restrictions, in SQL Server Standard.

2.3 Mirroring

Database mirroring is a software solution to increase database availability by supporting almost instantaneous failover. To configure database mirroring we need two servers, principle server, mirror server and optionally witness server. Mirroring can be configured from the principle server, which is the production server. To configure mirroring we need to right click on database and follow the database-mirroring wizard.

Each database-mirroring configuration involves a principal server that contains the principal database, and a mirror server that contains the mirror database. The mirror server continuously brings the mirror database up to date with the principal database.

Prerequisites for configuring database mirroring are both the principle and mirror server should be in full recovery mode:

We can check the mode by executing following query:

Select name, recovery_model_desc from sys.databases where db_name = ‘ DBNAME”

Now we need to take full database backup and transactional log backup of database under consideration on primary server and restore it on mirror server.

This can be done with following set of commands:

Full Backup: Backup Database Database_Name TO Disk = “D:\DBNAME.bak”;

Transactional Backup: Backup LOG Database_Name TO Disk = “ D:\DBNAME.trn”;

Once backup is complete we can copy paste backup file on mirror server across the network.

Now we need to restore the backup.

Database mirroring runs in two modes synchronous operation in high-safety mode, or asynchronous operation in high-performance mode. In high-performance mode, the transactions commit without waiting for the mirror server to write the log to disk, which maximizes performance. In high-safety mode, a committed transaction is committed on both partners, but at the risk of increased transaction latency.

In its simplest configuration, database mirroring involves only the principal and mirror servers. In this configuration, if the principal server is lost, the mirror server can be used as a warm standby server, with possible data loss. High-safety mode supports an alternative configuration, high-safety mode with automatic failover. This configuration involves a third server instance, known as a witness, which enables the mirror server to act as a hot standby server. Failover from the principal database to the mirror database typically takes several seconds.

2.4 Log Shipping

Log shipping operates at the database level. You can use log shipping to create warm standby databases for a production database. Production Database is referred to as Primary Database. Standby databases are referred to as secondary databases. Each secondary database is created by restoring a database backup of the primary database with no recovery, or with standby. Secondary database can be used for reporting purpose.

A log shipping configuration includes a single primary server that contains the primary database, one or more secondary servers that each have a secondary database, and a monitor server. Each secondary server updates its secondary database at set intervals from log backups of the primary database. There are three jobs basically which runs at primary database namely backup job, copy job and restore job. Backup job take backup of the transactions, Copy job ship those transactions where as restore job restore the transactions on secondary database.

Log shipping provides the flexibility of supporting multiple standby databases. If you require multiple standby databases, you can use log shipping alone or as a supplement to database mirroring. When these solutions are used together, the current principal database of the database-mirroring configuration is also the current primary database of the log-shipping configuration.

Log shipping is supported in the SQL Server Enterprise, Standard, and Workgroup editions.

2.5 Replication

Replication uses a publish-subscribe model. This lets a primary server, referred to as Publisher; distribute data to one or more secondary servers, known as subscribers. Replication enables real time availability. It supports filtering to provide a subset of data at Subscribers, and also allows for partitioned updates. Subscribers are online and available for reporting or other functions, without query recovery. SQL Server offers three types of replication: snapshot, transactional, and merge. Transactional replication provides the lowest latency and is usually used for high availability. Replication is supported in all editions of SQL Server.

3. One should take into account following considerations before selecting a high-availability solution

3.1 Failover clustering and database mirroring both provide the following:

- It provides Automatic detection and failover
- Manual failover is possible.
- Transparent client redirect

Failover clustering has the following constraints:

- It operates at the server instance scope
- It requires signed hardware
- It has no reporting on standby
- It utilizes a single copy of the database
- It does not protect against disk failure

3.2 Database mirroring offers the following benefits:

- It operates at the database scope.
- It uses a single, duplicate copy of the database
- It provides limited reporting on the mirror server by using database snapshots.
- When it operates synchronously, provides for zero work loss through delayed commit on the principal database.
- Database mirroring offers a substantive increase in availability over the level previously possible with SQL Server and offer an easy-to-manage alternative to failover clustering.

3.3 Log shipping

- It supports multiple secondary databases on multiple server instances for a single primary database.
- It allows a user-specified delay between when the primary server backs up the log of the primary database and when the secondary servers must restore the log backup. A longer delay can be useful, for example, if data is accidentally changed on the primary database. If the accidental change is noticed quickly, a delay can let you retrieve still unchanged data from a secondary database before the change is reflected there.
- Asynchronous database mirroring has the potential advantage over log shipping of a shorter time between when a given change is made in the primary database and when that change is reflected to the mirror database.

- An advantage of database mirroring over log shipping is that high-safety mode is a no data loss configuration that is supported as a simple failover strategy.

3.4 Replication

Replication offers the following benefits:

- It allows filtering in the database to provide a subset of data at the secondary databases because it operates at the database scope
- It allows more than one redundant copy of the database
- It allows real-time availability and scalability across multiple databases, supporting partitioned updates
- Allows complete availability of the secondary databases for reporting or other functions, without query recovery.

Conclusion

Now depending upon the business requirements, one can offer the most suitable solution to the client. Client needs should be taken into account before implementing high availability solutions. There are possibly two scenarios i.e. Client can afford downtime but he is not willing to invest that m heavily so the suggested solution will not offer minimum downtime and there will be some data loss but it is cost effective. This mostly suits to Testing or Development environment. Second scenario is when client is managing a critical application say financial application and cannot afford downtime or data loss at any cost. Therefore, now the suggested solution will ensure no downtime but is not cost effective. In this case MS SQL Server 2012 AlwaysOn is a best option to choose.

References:

- High Availability and Disaster Recovery (OLTP)---a Technical Reference Guide for Designing Mission-Critical OLTP Solutions
- High Availability Solutions (SQL Server)
- High Availability and Disaster Recovery at Service: A SQL Server 2008 Technical Case Study
- Microsoft SQL Server AlwaysOn Solutions Guide for High Availability and Disaster Recover
- Quorum: How a Witness Affects Database Availability (Database Mirroring)
- Quorum vote configuration check in AlwaysOn Availability Group Wizards (Andy Jing)
- WSFC Quorum Modes and Voting Configuration (SQL Server)
- Cross-Database Transactions Not Supported For Database Mirroring or AlwaysOn Availability Groups (SQL Server)
- Database Mirroring (OLTP)---a Technical Reference Guide for Designing Mission-Critical OLTP

