

SQL SERVER DBA

SQL SERVER DBA CONTENTS

F DBA ROLES AND RESPONSIBILITIES - 1

SQL SERVER ARCHITECTURE - 3

- Storage Architecture
- Database Architecture
- Transaction log Architecture

INSTALLATION - 21

- Pre-requisites
- Installing SQL Server 2012
- Post Installation steps.

DATA BASE - 33

- System databases
- Creating databases

BACKUPS - 41

- Recovery Models
- Backup types
- Advanced Backups

RESTORATION - 55

- Pre restoration
- Post restoration
- Restore enhancements in SQL 2012

SECURITY - 67

AUTOMATION - 79

MIGRATION - 85

UPGRADATION - 93

LOGSHIPPING - 97

- Log shipping Architecture
- Components of Logshipping
- Failover steps

DATABASE MIRRORING - 107

- Components in Mirroring
- How Mirroring Works
- Mirroring situations

REPLICATION - 115

- How replication works
- Configuring Replication

CLUSTERING - 129

- Clustering Architecture
- Components
- Clustering Admin tasks.
- Configuring Windows / SQL clustering

ALWAYSON - 141

PERFORMANCE TUNING - 147

- Locks, blocking and Deadlocks
- CPU, Memory, Io issues trouble shooting
- SQL Profiler
- Indexes and Query Tuning

DATABASE ADMINISTRATION

Database Administrator is a person responsible for database design, Implementation, Maintenance and repair of the database.

The Main goals of DBA is to keep the database server always up and make it available to users. In case of any failures DBA should minimize the data loss by implementing powerful backup and restoring Techniques.

DBA Responsibilities

As a DBA we have to perform these.

1. Maintaining the availability of databases by minimizing the down time.
2. Data recovery, We have to minimize the data loss in case of failures by implementing High availabilities.
3. Provide high security in accessing the databases externally.
4. Need to monitor the performance of servers, Implement various techniques to increase the Performance
5. Regularly Monitor database growth, diskspace, Sql server logs, Event viewer logs to avoid issues and to identify any bottlenecks.

DBA Roles / Daily Activities

1. As part of DBA team need to provide 24/7 production support to client and users.
2. Work on user requirements and problems which comes in the form of tickets.
3. Responding to alerts which we receive in the form of email from Third party monitoring tools.
4. Make sure all the maintenance jobs are running successfully.
5. Make sure all backup jobs executed successfully with out any issues on all servers.
6. Checking sqlserver logs to identify bottlenecks.
7. Checking drive spaces on critical servers to ensure that there is ample amount of space
8. Regularly monitor datafile growth, log file growth database growth as part of capacity planning.
9. Maintain documentation of all the tasks and issues that you encounter for future reference.
10. check whether all sql services are running (or) not.

SQL SERVER ARCHITECTURE

SQL server follows client-server architecture. When ever user performs any action on client machine, it converts in the form of query. This query moves from client to server in the form of Network packets using Protocols for connection and communication between source and destination servers.

SQL server is mainly divided into 2 Engines

1. Relational Engine
2. Storage Engine

Relational Engine (Query processor) prepares execution plan and handover to storage engine.

Storage Engine is a central repository, responsible in execution of query using execution plan, response sent to user.

Buffer pool is another important component contains Plan cache and data cache which is used for query execution.

SQLOS is core to SQL server architecture, used for scheduling, I/O completion, Memory management and resource management. It is a thin layer between Windows OS and SQL Server.

Components of SQL Server

SQL Server Network Interface (SNI) :

SNI is a protocol layer that establishes the network connection between the client and server. It uses TCP/IP Protocol to send queries in the form of TDS packets.

Command parser

Command parser first checks for syntax errors, then it generates query plan or find an existing plan. Query plan contains detail steps how query is going to execute. Command parser checks whether a plan already exists in plan cache of Bufferpool. If finds plan passes to query executor for execution. If it does not find then query passes to optimizer.

Optimizer

Optimizer prepares query plans for one query in that SQL server select best plan based on response time, the query plan passes to query executor for execution.

Query Executor

Query executor requires data to read the query plan it passes to Access Methods of storage engine.

Access Methods

Access methods requires data to complete the query. It asks buffer manager to provide data page. Once it receives required data, the query results passes back to Relational Engine and then to user.

Buffer Manager

Buffer Manager checks in data cache of buffer pool to see if it has page already in cache memory. If page exists, it passes results to Access methods. If not exists it pulls required pages from mdf datafile. Put in data cache and passed it back to Access methods.

Plan Cache

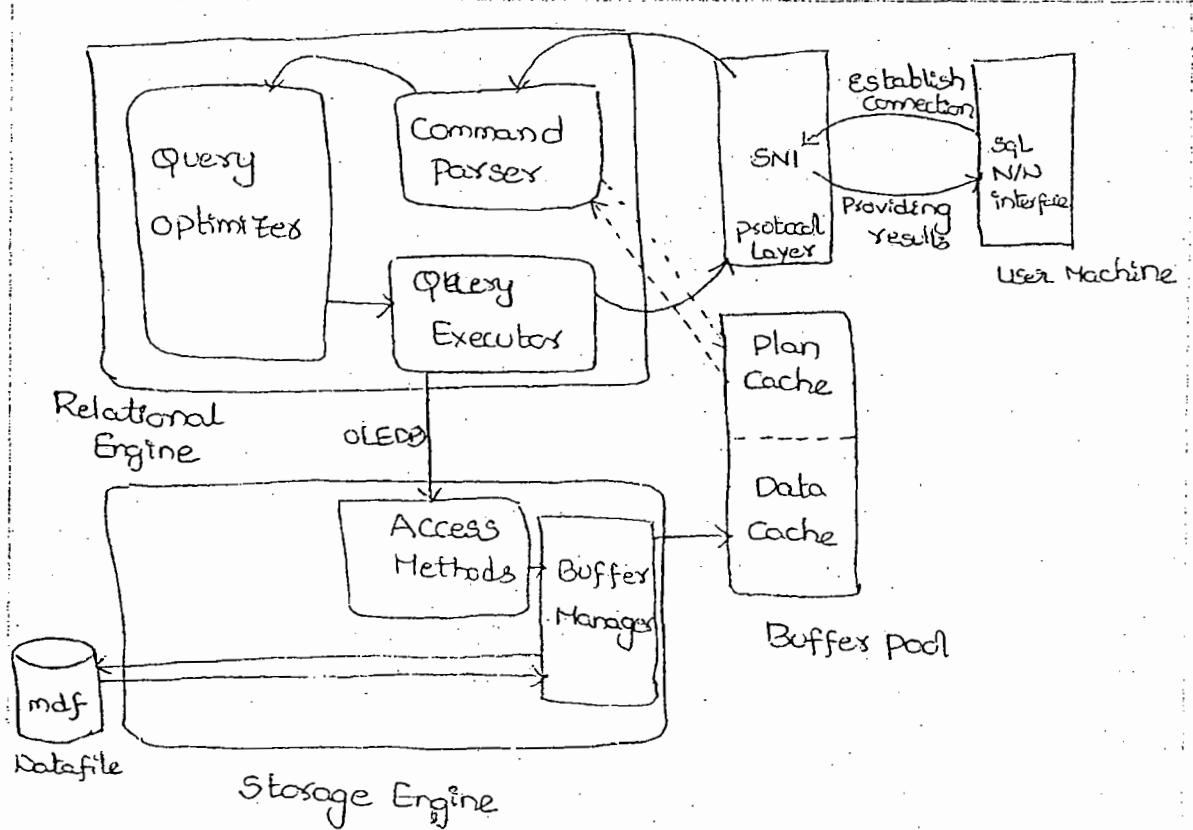
Part of SQL Server's buffer pool used to store previously executed execution plans in case they are needed later.

Data Cache

Data Cache is largest part of buffer pool. Every data page that is read from disk is written a copy here before using. Under memory pressure these pages are flushed from cache using LRU (Least Recently Used).

Policy

Sql Server architecture



Steps in executing a query.

1. SNI of user establishes connection between client and server using TCP/IP protocol, sends query in TDS packets.
2. Query at Command parser checks syntax errors then checks plan in plan cache of bufferpool. If plan not exists, passes the query to optimizer.
3. Optimizer generates best plan and pass to query executor, it reads the plan and passes to Access Method of storage engine through OLEDD.
4. Access method requests Buffer Manager to

- E. Provide the data.
- F. 5. Buffer Manager checks in data cache of buffer pool for existing page. If page not exists it pulls the required pages from Data(mdf) file, put in data cache and pass to access method.
- 6. Finally Access Methods passes the results back to relational engine, from there it sent back to user who executed the query.

Protocols available in SQL Server

SQL Server Network interface (SNI) is a protocol layer that establishes the network connection between the client and server. SQL Server supports 4 protocols

1. Shared Memory
2. Named Pipes
3. TCP/IP
4. VIA

Shared Memory

It is default protocol used to connect client and SQL Server on the same machine.

Named Pipes

Client and server will connect with in a LAN.
It has certain limitation.

TCP/IP

TCP/IP is most used protocol for SQL Server client establishes connection with SQL Server using our IP address and a port number, 1433. We can access the databases using internet hence there are no boundaries for this protocol.

VIA (Virtual Interface Adapter)

VIA is a wireless intranet protocol for connecting client and server within a certain range.

To establish a secure SQL connection we need a port number along with protocol.

Default port number for TCP/IP Protocol is 1433.

We can change the port number from

Configuration Manager → SQL Server Network Configuration

→ Protocols → We can change.

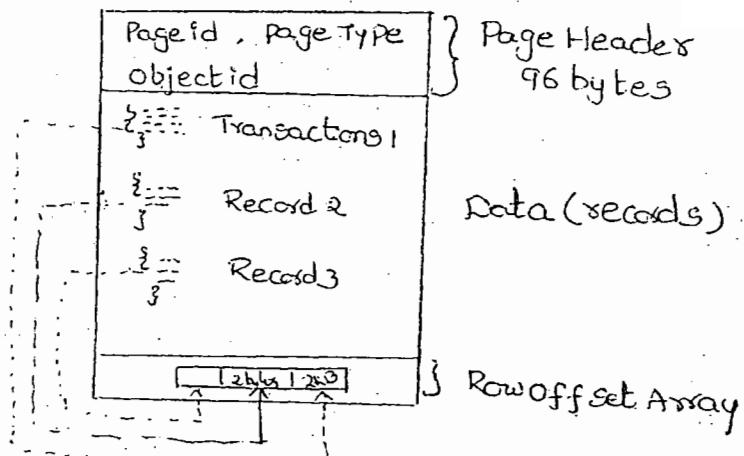
STORAGE ARCHITECTURE

In SQL Server all the data will be stored in the form of records, these records also called as row data.

All these records further grouped into a Page.

Page is a default storage unit of SQL Server. The size of Page is 8kb.

Page Architecture



Page consists of 3 sections

- ① Page Header
- ② Actual data
- ③ Row offset array.

Page Header consists of Page id, Page type, Object id, Header version.

Page id → To identify particular page using unique page id.

Page type → What type of page it is either Data page or Index page.

In Row offset location of record will be stored (2bytes)

Types of Pages

1. Data page → stores data entered by user.
2. Index page → indexes are pointers which store address of original pages for quickly locating data.
3. Free space page → It stores Page allocation information and Unused space available on pages.
4. Text/Image → It stores Large object data (LOB) like Text, image and XML data.
5. GAM (Global Allocation Map) & SGAM (Shared Global Allocation Map) → It stores Extents allocation information.
6. BCM (Bulk changed Map) → stores Extents information in a Bulk operation.
7. DCM (Differential change Map) → It stores modified Extents information after Full back up.
8. IAM (Index Allocation Map) → stores Extents information that are used by a table (or) index.

These are important types of Pages. All these Pages are further grouped into a Extent.

Extents

Extent is a storage structure consists of 8 consecutive Sql server pages. Pages in a Extent can be owned by one table (or) upto Eight tables.

There are 2 types of Extents

1. Uniform Extent :

If all pages are going to store same table data.

2. Mixed Extent :

If the pages shared by 2 (or) more tables.

When a table is created and a row is inserted

Table gets 1 Page in Mixed Extent, when a table grows then these tables moved to uniform extent.

This is to manage space efficiently.

All the extents further group into a File. on a file we will have better control in sql server

There are 2 types of files mainly

1. MDF (Master data file)

2. LDF (Log data file)

Mdf stores permanent data

Ldf stores changes information will be recorded

Later these changes apply on mdf data.

Database

Files combine to form database. We require minimum 2 files 1 Mdf and 1 Ldf to create a database.

Maximum we can 'n' number of files means no limit.

Filegroups

Some files stores system data and some store user database data. Logically dividing databases into groups called Filegroups.

Storage structure

Records (rowdata)



Pages



Extents



Files (Mdf, Ldf)



Database

DATABASE ARCHITECTURE

SQL Server stores data mainly in 2 types of files

1. Data file (MDF)

2. Log file (LDF)

Data file stores actual data with .mdf extension.

It stores permanent data.

Log files stores modified recorded information with

.ldf extension.

We have another file called secondary data file

.ndf extension. A database may (or) may not have these secondary data files

SQL SERVER TRANSACTIONS

A transaction is a set of T-SQL statements that read and write data into the database.

There are 2 types of Transactions.

Implicit and Explicit.

Implicit transactions are those without Begin Tran.

Explicit transactions are started using Begin Transaction and are controlled by using T-SQL Command Commit transaction (or) Rollback transaction.

Any transaction should process ACID Properties then only changes move from Ldf to mdf

ACID Properties

Atomicity, Consistency, Isolation and Durability.

Atomicity

Atomicity means all the statements of a transaction must complete successfully or rolled back completely. Means either all (or) None to updated.

Consistency

Consistency means a transaction never leaves database in half-finished state, when ever any change happen on Parent object it should automatically reflect on dependent child object to ensure that database in a consistent state.

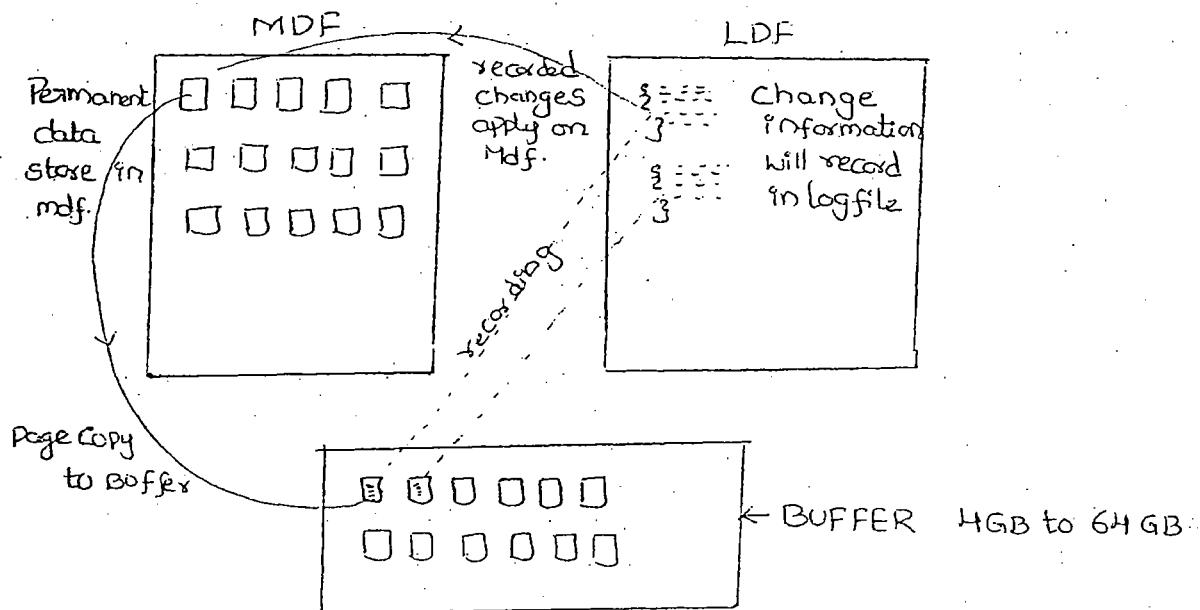
Isolation

Isolation keeps changes of incomplete transactions independent from one another.

Durability

Once a transaction is committed, it must be permanent even if there is a system failure means it cannot be rolled back.

Data base Architecture



MDF contains Permanent data

LDF contains whatever changes we are performing on database all the change related information will be recorded in Ldf file.

Buffer is a ram to perform modifications on a copy of Permanent Page. once it commits record the information will record in Ldf and same changes apply on mdf when check point runs.

How it Works

Sqlserver will not allow to do modifications directly on Mdf. sqlserver will make a copy of Pages from Mdf to buffer. once transaction is full committed it records the information that what type of

data he is inserting, Number of Pages effecting what he is performing all these change related information will record in same sequential way in Log file. Pages will stay some time in buffer for faster retrieval. Read and write operations from buffer will be very faster comparing to operations from mdf data. Using recorded information when ever check point runs on log file, it applies same changes permanently on Mdf file.

Check point process

Check point is a internal mechanism performs regularly based on number of transactions or number of pages, there is no time interval for running this. Checkpoint scans log file, checks how many committed transactions are there, how many failed and how many still running. Committed transactions move to Mdf, failed transactions will be rolled back. Currently running transactions will not to be touched by check point.

Advantages of check point

1. Checkpoint help in speeding up recovery process.
2. Checkpoint helping in committing data permanently.

Transaction Log Architecture

SQL Server uses LSN (Log Sequential number) in identifying the transaction. Each and every transaction that comes to log file will associate with a LSN number. Roll-forward and roll-back will be done internally using these LSN numbers only.

WAL (Write Ahead Logging)

Before commit in mdf every transaction should written a entry in log file is called WAL. Transactions never comes to mdf directly.

Log file divided in to 2 parts

1. Active portion (or) physical log
2. Inactive portion (or) virtual log.

Active log portion

When user performs transactions it will have 3 states

1. Committed in log file and waiting for checkpoint.
2. failed in the middle.
3. Transactions still running

All these 3 state transactions will be in Active portion of log file. When check point runs committed transactions make a copy in Inactive portion and moves to mdf.

Inactive log

SQL Server maintains fully committed transaction in these

Recovery Process in SQL Server

When ever SQL Server restarts check point verifies pending transactions before restart, SQL Server will perform recovery process. This process will analyze what is the state of log file and perform 2 operations.

1. Redo (or) Roll forward → Committed changes will be moved from Ldf to mdf permanently
2. Undo (or) Roll back → failed transactions and running transactions will be deleted from log file.

Once this recovery process complete then only users can able to access the database.

Lazy Writer

Modified pages will be in buffer some time, When ever buffer is about to fill with these modified pages,

Lazy writer is another internal mechanism usually in sleep mode invokes and clear the buffer pages.

It uses LRU algorithm in clearing, LRU stands for Least Recently used pages, on page header of page there will be reference counter means how many times this page being used, based on counter least used pages will be deleted in buffer.

Dirty Pages :- Pages committed in log file and waiting for check point to move mdf, those called dirty pages.

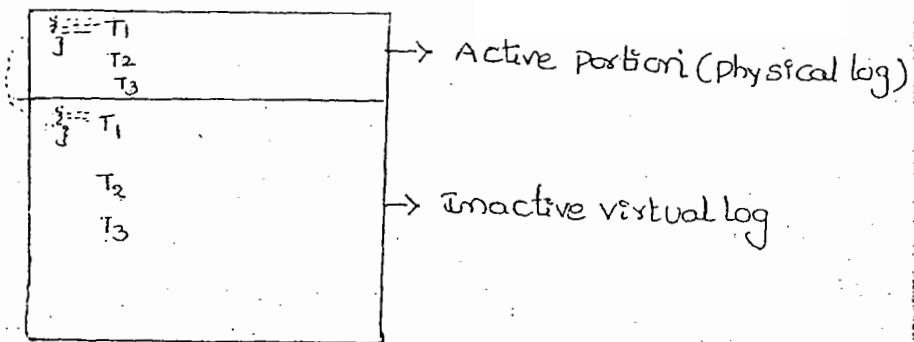
Inactive portion. This portion only used for taking the backup of log: whenever we take log backup it copies the inactive portion and truncates the inactive portion. We have 2 types of backup for log portions.

Full backup takes backup of Mdf and Active log portion

Log backup takes backup of Inactive log portion

This portion we call as virtual log. SQL Server not uses these records that's why it call as Inactive virtual logs.

Log Architecture



Inactive portion further divided into more virtual logs. We have a property called Log reusability. Log backup copy inactive portion to a file and truncates the log data. Same space can be used multiple times called Log reusability concept.

Transaction log is a cyclic process of writing log record into virtual log file by SQL Server. Whenever one virtual log is filled up it will goes to next virtual log. If all virtual logs files are filled up the inactive

Position will grow further and creates more virtual logs, till we have log space allocated. If it cannot grow further it will throw an error " Transaction log for database is full and transaction will fail".

The only way to clear inactive virtual log is to take log backup it truncates logs. After truncation this space will be released. Backup will not affect position.

Advantages of T-log

1. It provides Transactional consistency
2. It provides Transactional recoverability.
3. It provides Log reusability.

Enhancement in sqlserver 2012 (Indirect checkpoint)

Check point occurs automatically based on workload (or) by a certain operations internally. We can set sql server level recovery interval to run checkpoint by using SP-configure (or) Manually issue a checkpoint using check point Tsql command.

In sqlserver 2012, we have enhancement on checkpoint at database level using TARGET-RECOVERY-TIME option. Increasing time of recovery to seconds (or) minutes.

Alter database sqltest2012 set Target_recovery_time = 5seconds

Here Every 5 seconds checkpoint will occur.

INSTALLATION

Whenever we want to start installation of sqlserver
we need to gather details of installation.

Version of sqlserver.

Version (Name)	RTM	SPI	SP2	SP3	SP4
SQL Server 2014 (Hekaton)	Trial version released				
SQL Server 2012 (Denali)	11.00.2100.60	11.00.3000			
SQL Server 2008 R2 (Kilimanjaro)	10.50.1600	10.50.2500	10.50.4000	10.50.5000	
SQL Server 2008 (Katmai)	10.00	10.00.253	10.00.4000	10.00.5800	
SQL Server 2005 (Yukon)	9.00.1399.06	9.00.2047	9.00.3042	9.00.4035	9.00.5000
SQL Server 2000 (Shiloh)	8.00.194	8.00.384	8.00.532	8.00.760	8.00.2039
SQL Server 7.0 (Sphinx)	7.00.623	7.00.699	7.00.842	7.00.961	7.00.1063

Editions of sqlserver

Editions available in sql server

1. Free editions
2. Special editions
3. Core editions

Express and Evaluation edition falls under free editions

One is free of cost and evaluation expires after 180 days.

No need to buy licenses from Microsoft.

Special editions are

Web edition for Web developers, developer edition for [21]
designing stored proc and exclusively for developing purpose.

Workgroup edition for small scale industries

We do not use free and special editions in production servers. We use mainly Enterprise edition.

Core Production Editions

1. Standard edition

OS Maximum → No limitation from SQL Server side, depends on what operating system supports. Advanced features like online restoration, encryption not available.

2. Enterprise edition.

King of all SQL Server editions, No limitation from SQL Server. Cost approximately 3 times more than standard. Mostly used edition in production environments.

3. Business Intelligence edition

Business Intelligence is new edition released in SQL Server 2012 by Microsoft. This edition consists of all standard features along with additional reporting features for Business Intelligence (BI).

Windows operating system editions

1. Server edition
2. Advanced edition
3. Enterprise edition
4. Data center.

Windows Admin will perform this installation and provides the server to SQL DBA to perform SQL installation. Microsoft recommends to get complete enhanced features of SQL Server use same windows operating system.

Ex:- SQL Server 2012 provides best performance with Windows Server 2012. As most SQL Server components integrated with windows operating only.

Components to be selected

SQL Server mainly depends on services.

1. Database services

Under this service database engine will run.

2. Integration Services (SSIS)

From multiple sources ETL (Extract Transform Load) will integrate data.

3. Reporting services (SSRS)

To generate reports very quickly.

4. Analysis services (SSAS)

For notification purpose.

Instance

Instance is Name of Server, It is a unique identifier.

For the first time if we execute SQL Server setup, the installation uses the Windows registered name that we called as default instance.

Named Instances, If we run SQL Server setup again a new SQL Server will be created to work. Each and every time it creates multiple instances.

only one default instance will be there on the server.

Named instance depends on SQL Server. In SQL Server 2012 it supports 16 instances.

When ever we install SQL Server it will create new binary files for each installation. Shared components like Management studio, Business intelligence studio and other features will install only once.

Binary file location

By default SQL Server installation creates set of files in C:\Program files\Microsoft\SQLServer. If we want to change the path we can keep them on E(8)G drive.

Collation settings

SQL Server by default uses character set and sort order. Common keys on the keyboard that is going to support. By default it uses.

Authentication

Process of validating login account is authentication.

We have 2 types ① Windows ② Mixed mode.

Windows Authentication

It allows only windows logins which are stored in Active directory. SQL Server uses same account for connecting.

Mixed mode Authentication

It allows window login and SQL Server login.

When ever login tries to connect first it validates in Active directory if not found then comes to syslogins for validating if found establishes connection.

Service Accounts

Services used for better control on SQL Server. One program may have multiple services. Entire SQL Server runs using these services. We have

1. SQL Server service

2. SQL Agent service

3. SQL browser service

4. Full Text Service

5. MSDTC Service.

SQL Server Service

If this service is running, SQL Server will run. Most important service. All other 4 services depend on this service. DB engine runs under influence of SQL Server service.

SQL Agent Service

Entire Automation process runs under this service. Agent service will look at scheduled time of jobs and invoke tasks automatically.

SQL Browser Service

In Production service this service will be disabled due to security reasons.

Full Text Service

We keep most repeated words in catalog for faster search of data from database.

MSDTC

Microsoft distribution transaction coordinator useful for handling distributed transactions from other server.

Installing Sqlserver 2012

Before installing sqlserver , server needs the following.

Prerequisites

- Net frame Work 4.0
- Windows Power shell 2.0
- Net 3.5 with Service Pack 1

Sqlserver 2012 installation

Sqlserver 2012 installation is almost same as sqlserver 2008.

Menu of sqlserver installation center.

Planning

Installation

Maintenance

Tools

Resources

Advanced

Options

Planning helps in reviewing

1. Hard ware and software requirements.
2. Security considerations.
3. Online release notes
4. Install upgrade Advisor
5. How to get started with sqlserver 2012 failover clustering

Planning varies hardware and software requirements of the server. System configuration checker (SCC) scans the machine where installation of SQL Server exists and provide results that to avoid obstacles while installing. Upgrade Advisor will help while upgrading.

Installation

Installation	New SQL Server installation (or) Add features New SQL Server failover cluster installation Add node to SQL Server failover cluster Upgrade from SQL Server 2008 R2
--------------	---

Installation provides various installation options

- We can install new SQL Server for the first time (or) we can add additional features to an existing installation
- We can install clustering and add a node to the cluster
- We can upgrade to SQL Server 2012 from old versions.

Maintenance

Maintenance	Edition upgrade Repair Remove node from SQL Server failover cluster
-------------	---

We can upgrade and repair the SQL Server from here

Tools

Tools

System configuration checker

Installed SQL Server features discovery report

Upgrade Integration Services packages

Power pivot configuration tool

System configuration checker scans SQL Server installation.

Installed SQL features provides detailed report about SQL products installed on server.

Upgrade Integration Packages is a wizard steps in upgrading lower version SSIS packages to SQL Server 2012 packages.

Resources

Resources

SQL Server 2012 Books online

SQL Server Tech Center

Resources will help in knowledge on SQL Server 2012.

Advanced

Advanced

Install based on configuration file.

Advanced cluster preparation / completion.

Image Preparation of stand alone server

Configuration file install allows install configuration file on existing file.

Advanced cluster provides wizard for failover clustering.

Image Preparation gives install methods in SQL Server 2012.

In options, we can specify architecture of sqlserver install
or to change location of install files.

Once all these details provided we can start actual installation.

SQL Server 2012 set up

Set up support rules

Setup role

Features selection

Disk space requirements

Server configuration

Distributed Replay controller

Distributed replay client

Error reporting

Install configuration rules

Ready to Install

Installation Progress

Complete.

Installation process starts with system checks and validations.

Setup role is new screen in SQL Server 2012, to choose to install all features with default settings.

Feature selection → Provides all SQL components to select.

Install rules → Need to select default (or) named instance.

Disk space requirements → provide details whether we have specified space (or) not.

Server configuration → what are the services available in server.

Distribution replay controller and distributed replay client are new features in sqlserver 2012, allows us to replay our captured trace files of sqlserver.

Error reporting → provides feedbacks to Microsoft if want to send.

Install configuration rules →

Ready to install → provides all the features what we have selected

Installation progress → actual installation starts complete → completes sqlserver installation.

sqlserver 2012 installation completed

Post installation steps

1. Verify Protocols enabled (or) not

Microsoft SQL Server 2012 → SQL Server Configuration

Manager → SQL Server Network Configuration →

Protocols for MSSQLServer

2. Open services in SQL Server Configuration Manager

Select SQL Services and verify whether they are started (or) not.

3. From Run Command type services.msc and Verify the mssqlservices status and mode.

4. Verify the Advanced options using

sp_configure 'show advanced options', 1;

GO

Reconfigure

5. Specify Maximum and Minimum server Memory

sp_configure 'Max server memory (MB)', 8192;

sp_configure 'Min server Memory (MB)', 1024;

6. Enable Backup Compression, if we want

sp_configure 'Backup compression default', 1

GO

Reconfigure with override

GO

Data bases

Database :- It is a container which store data and data objects. It manages data and allows fast storage and retrieval of that data.

Types of databases

There are 2 types of databases.

1. System databases.
2. User databases.

System databases

System databases will be created as a part of SQL Server installation. SQL Server internal operations will be performed using system databases.

User databases are created based on organisation requirement. We are creating externally and stores data.

System databases available in SQL Server from SQL 2000

1. Master
2. Model
3. Msdb
4. Tempdb
5. Distribution db (only available while replication configuring)

From SQL Server 2005 along with Master, Model, Msdb and Tempdb a new database introduced called Resource db

Master

The first database loaded by sqlserver on services start is Master db. Once master db starts then only other databases will start. It is just like a brain to sqlserver.

→ Master db stores critical information like sqlserver configuration such as CPU info, memory configurations, SP-configure settings.

→ Master stores other system databases information in 'sys.databases' table. If it not start other databases will not start.

→ In Master db we have 'sysaltfiles' table which contains location of all system databases mdf and ldf will be stored. To start any database it has to come to Master and then starts database.

→ SQL logins will be stored in 'sys.logins' table of Masterdb. If any user connected to sqlserver externally his login will be authenticated in Masterdb.

→ It stores other server objects. Linked servers, endpoint etc. Till sql2000 we store all meta data (data about data) in tables. If any binary file effect entire data will be lost. Hence from 2005 we are storing metadata in views.

Model

Model database will act as a Template ...
New user databases. How many files we want to create
How many mdf and ldf, what is initial size, Path
of binary files all these information we call as
a Template. Every New database acquire properties
from Model database. New db goes to Model database
and get those properties. If we create any table
all the new databases which we create from now on
contains same table.

Msdb

Msdb stores all scheduling and Automation information.
In SQL Server we are doing Automation in the form of
job; SQL Server Agent service will be responsible for this.
→ All SQL jobs information will be stored in sysjobs,
syschedules and syssteps tables inside Msdb.
→ SQL Server Agent service will read Msdb schedule
tables, based on this it will start and stop the
tasks automatically.
→ History related information like Backup, restoration
Logshipping history will be stored.

Temp db

Temporary database where all temporary objects will be created for performing sqlserver internal operations.

- Temporary objects that are created by sql server internal operations such as temporary tables, temporary stored procedures etc,
- Rows versions that are generated by data modifications transaction in database that uses read-committed (or) Snapshot isolation stores here.
- All the transformation, internal calculations, conditions sorting operations will be performed in Temp db.

When ever sqlserver started Temp db will be deleted and NewCopy will be created.

Resource db

Resource db is hidden database we can see only mdf and ldf physically. It stores where we keep sql binaries files.

- It provides high security to metadata, it contains all system objects such as sys.objects physically stores.
- Resource db makes upgradation to new sqlserver version easier and faster. In earlier versions upgrading will delete and create system objects. Now Resource database will contain all system objects. We take copy of Resource database mdf and ldf and completes upgrading easily.

Creating Databases

Open SQL Server 2012 → Connect to Management Studio

→ Select SQL Server instance → go to database rightclick

→ New database

General	Database name :	SQL test 2012
	Owner :	SA
	Logical Name :	sql test 2012-data, sql test-log
	Initial size :	1 GB
	Autogrowth :	500 MB
	Path :	E:\Database\sql test.mdf · ldf

Database name, Provide name to database

Owner shows default, if we want we can change.

Logical name is used to refer the physical file in all T-SQL statements.

Initial size will acquire from Model db properties we can change as per our requirement. What is the size of database while creating.

Autogrowth means once it reaches the initial how much my database size can increase. It will add size.

Path, actual storage location of database, where my database is going to store (location).

Database Properties

We require Minimum 2 files to create a database
1 mdf and 1 ldf.

These are 5 properties required to create a database

1. Name
2. Filename (Path)
3. Initial size
4. Max size
5. File growth

Name, what is the database name that we are going to provide for new user database. Logical mdf and logical ldf will use for Administrative purpose.

Filename, complete path where database is going to store.

Initial size, we have estimate how much the database size initially we need. We have to specify the Max limit and Autogrowth means once it reaches the initial size automatically how much size it has to increase. We can provide it either in MB (or) Percentage.

To create database using T-SQL

Create database sqldb2012;

It will create new database with default model properties

We can specify the properties through T-SQL

Create database sqldb2012

on

(Name = sqldb2012_data

filename = 'E:\data1\sqldb2012-data.mdf'

size = 10GB

MaxSize = 500GB

filegrowth = 5GB

)

Log on

(Name = 'sqldb2012-Log'

filename = 'E:\data1\sqldb2012-log.ldf'

size = 1GB

MaxSize = 50GB

filegrowth = 500MB

)

Rename a filename

Alter database sqldb2012 modify file

name = sqldb2012- filename = 'D:\sqldb-data.mdf'

When database is online we cannot move mdf and ldf. It may lead to database crash.

After creating database, verify Properties

Files tab → shows how many Mdf and Ldf are there

In database we have Add option in GUI we can add more files.

Filegroup → By default we have only 1 filegroup as primary. We can group multiple files under one filegroup

options

Collation setting → Latin1-General

Compatibility level → on which version sql server developed show here

For Sqlserver 2012 it is 110

Back UPS

Backup means exact copy of database.

Situations where backup can use

1. Hardware (or) software failure
2. Natural disasters
3. Internal misuse.

Hardware failure includes operating system's CPU memory, network card failures.

Software failure includes operating system failures, database server failures. We have one more failure called storage failure.

Natural disasters include natural calamities like cyclones, fire attacks.

Internal misuse means some one in the organization wantedly (or) by mistake delete some data.

Backup objectives

1. To recover lost data.
2. To minimize the data loss
3. To minimize the downtime.

Recovery Models

Understanding the recovery models is essential to develop an effective backup strategy.

A database recovery model determines how transactions are logged and type of backups that can be performed on a database.

There are 3 recovery Models

1. Simple recovery model
2. Bulk-logged recovery
3. Full recovery.

Simple recovery

Simple recovery supports full and differential backups.

"Truncate log on" checkpoint option will be enabled in simple recovery, whenever checkpoint runs in this model. Committed data moves to Mdf and inactive portion of virtual log file will be truncated, hence there will be no data in Ldf file for backup's.

Point-in-time restoration and point of failure when database crash is not possible.

This model is used for development and test servers where data loss is acceptable.

Full recovery

Full recovery provides highest level of data protection by fully logging all transactions. It supports all type of backups. We can recover and restore data fully. It is capable of point-in-time recovery.

Bulk-logged

When ever we perform bulk operations log file may full due to huge transactions. No new transactions will comes to mdf hence the transactions may fail.

In bulk logged recovery model it records minimal information about transactions. It logs the extent allocations and modified extents information only. We cannot perform Point-in-time recovery. When ever transactions including BCP, Bulk insert, Create Index, Alter index rebuild, DBCC DBREINDEX, SQL Server identifies it is a bulk operations and start logging data minimally.

When ever we change recovery Model always take a full backup to avoid data loss. Steps to be followed

- ① Change to Bulk-logged ② Allow transactions to complete
- ② Change to Full and take full backup immediately.

Backup Types

There are 3 basic types of backups

1. Full Backup
2. Differential Backup
3. Transaction Log Backup.

Full Backup

A full backup captures entire database including active part of transaction log. Any uncommitted transactions at the time of backup taking will be rolled back while restoring. For all backups full backup is the base.

SQL Server only backs up the data pages in the database excludes all empty data pages.

T-SQL Command to take full backup

```
Backup Database DatabaseName to disk = 'E:\backups  
test-Full.bak'
```

Differential Backup

Differential backup captures modified pages after the last full backup. Differential backup are incremental backups.

Differential backup uses Differential change map (DCM) page to record the changes on Extents.

File backups

File backups copies pages from primary and secondary files (or) file groups. This backup can be used to speed up restoration process. If we have multiple filegroup backup on separate disk. If single disk fails we can restore that file backup of single disk instead entire database.

T-SQL command for file back

```
Backup database sqltest2012 file = 'sql2012test_Data'  
to disk = 'E:\Backups\sqltest2012 - data.bak';
```

For Primary filegroup

```
Backup database sqltest2012 filegroup = 'Primary'  
to disk = 'E:\Backups\sqltest2012 - Primary.bak';
```

Partial Backup

Partial backups backup primary filegroup and every read and write filegroup that is part of database.

Partial backup use Read-Write-Filegroups option in the Backup command.

```
Backup database sqltest2012 Read-Write-Filegroups  
to disk = 'E:\Backup\sqltest2012.bak';
```

All the extents are marked with 0 inside DCM page.

For each modified page the extent will convert to 1.

SQL Server copies pages from mdf to buffer and converts value to 1 in buffer. Differential back up captures all the modified pages by using these converted 1's after full backup.

In differential first time all modified pages will be backed up. In the next differential it captures latest modified pages along with first differential modified pages. This process will continue till next full backup, hence it is called Incremental Backup.

T-SQL command

Backup database SQLtest2012 to disk =

'E:\backup\test-differential.bak' with Differential

T-Log backup

T-Log back up copies the inactive portion of the log and truncates the inactive portion. Once it truncates log space can be reused by new transactions.

T-Log backup are sequential backup's. Every backup will have LSN (log sequential number).

Backup Log SQLtest2012 to disk = 'E:\backup\Test-log.tbn'

Taking backup using SSMS GUI

1. Connect to sql server data through ...
2. Right click on database → From Menu → select Tasks
→ Backup
3. Select the type of backup and provide destination Path. General Tab provide below details.

General	Database	sqltest2012
	Recovery Model	Full
	Backup type	Full
	Backup Component	<input checked="" type="radio"/> Database <input type="radio"/> Filegroups
	Backup set	
	Name :	sql test 2012 Full Database backup
	Backup set will expire	<input checked="" type="radio"/> After <input type="text"/> 7 Days
	Destination	<input checked="" type="radio"/> Disk <input type="radio"/> Tape
	Add	Remove
	OK	Cancel

Select the database from dropdown, recovery model always be full, select database option in backup component. Backup set group all files in single location. Specify Retention period of back in days (or) provide exact date.

Using Add button provide backup filename for the backup device. If we want we can remove.

options overNrite media

- Backup to existing media set Append
- OverNrite all existing backup set

Reliability

- Verify backup when finished
- Perform check sum before writing
- Continue on error.

Compression

Set back up compression use default setting

Backup to existing media → Backup media will not over written

Append → Add backup files to end of existing files.

OverNrite → Overwrites existing backup files.

Verify back → Verifies whether data successfully copied
(or) not.

Check sum → checksum option verifies backup process
will test the page checksums that exist on data file.
If a bad page checksum is found, backup process will stop.

Continue on error → We are asking Sql Server to continue even though there is error in backup process.

Advanced types of Backup's

Copy only backup

Compressed backup

Split backup.

Tail log backup

Copy only backup

Copy only backup are used to perform a full (or) T-log backup without breaking the log (LSN) chain and without disturbing regular backup schedules. whenever we get adhoc request for backups we can use this copy only backup.

Through GUI

select copy only backup option , in backup window.

Through T-SQL Command

```
Backup database databaseName = E:\backups\sqltest
```

```
2012 - copyonly.bak' with copy_only.
```

From sqlserver 2008 we can able to take copy-only backup using GUI. In sqlserver 2005 we can only take using T-SQL command.

For Log

```
Backup Log : sqltest2012 to disk = 'E:\backups\sqllog.bnr'
```

```
with copy_only.
```

Compressed backup

From SQL Server 2008 this introduced. When we take backup it will compress the backup, it will not touch original database. It saves disk space and time. Once we restore it uncompress the backup to original size.

We can use SP-Configure stored procedure with 'backup compression default' 1 to enable and 0 to disable default compression for all backups in the server.

SP-Configure 'backup compression default', '1'
Reconfigure with override.

In the backup window select compression option at server level then compression will be enabled at server. Next time onwards for all backup it compresses the backup output file.

Set backup compression

Using T-SQL command:

Backup database Sqltest2012 to disk = 'E:\backups
\\Sqltest.bak' with compression = 1

'1' is to turn on compression, 0 is to turn off.

Split backup

When ever we have less space in drives we can split the backup into other drives which we gave space. It splits backup output in multiple files (drives). Performance will be very faster because more I/O of each drive will work hence backups will be faster.

We have one disadvantage while restoring if any split misses we can not restore entire database.

Backup database database name to disk

= 'E:\backups\sqltest1.bak'

to disk = 'F:\backups\sqltest2.bak'

to disk = 'G:\backups\sqltest3.bak'

Through GUI.

In backup window we specify more paths by adding E, F, G drives path for splitting.

Tail log backup

Tail Log backup refers the content of inactive portion of transaction log that has not been backed up.

It is just like normal T-log backup. This backup is always not possible to take.

Situations of Tail log backup

Whenever database is corrupted (or) damaged. If log file drive is still available, we keep database in Emergency state so that database will start with available files and drives. Emergency state makes database Read only and restrict the access to sysadmin only. Take tail log backup.

To keep database in Emergency state

Alter database sqltest2014 set Emergency.

Perform Tail log back immediately using NO-TRUNCATE Command to take

Backup log sqltest2014 to disk = 'E:\backups\

sqltest-Tail.trn' with NO-TRUNCATE

2. We can use No-Recovery option.

No Recovery means No one access the database until restoration complete. here we are preventing users to access the databases until restoration completes.

Backup log database name to disk = 'path' with No-recovery

In 'No-truncate' we are saying not to truncate the inactive portion until we complete the log back up in crash situation.

Backup options

Init / No init

Init stands for initialization, when we take backup

Init writes a record history of backup in Msdb database
but it overwrites the existing history

No init writes one new record under existing record.

Skip / No skip

Whenever we perform backup we can verify whether
it properly wrote a copy or not.

Skip will skip verification process

No skip will verify the verification process

Stats

Whenever we take backup for seeing status of that
backup Percentage completed. By using this option
we can estimate how much time will backup takes.

Backup database `databasename` to disk = 'Path' with stats

Backup history

We can see what are backups available in server
from backup history table of Msdb database

```
Select * from msdb.dbo.backupset
```

Backup Permissions

For taking backup we need certain permissions

1. System admin (sa)
2. Db_owner
3. Db_Creator
4. Db backup operator

We need any of the 4 roles for taking backup.

Backup history

We can get the details of the database who has taken backup, what type of backup it is, when this backup start and end, LSN's all these details we get by querying Msdb tables using query.

Select * from Msdb.dbo.backupset

D - database backup

I - Incremental/differential backup

L - Log backup

RESTORATION

F Restoration is a process of using backups to recover the exist database (or) to create a new database in another server.

Situations for restoration

1. Planned restoration
2. unplanned (or) Accidental restoration.

Planned restoration, When we have planned activities like Migration, upgradation, database refresh from prod to other servers, Configuring Logshipping and other High availabilities we use planned process in restoration. Here we will take fresh copy of backup and restore on the other server. Restorations are limited in this.

unplanned restoration

In Accidental restoration, Restoration depends on crash time and available backups at that time. Restorations are more here.

Restoration will be done in 3 steps

1. Pre restoration.
2. Actual restoration.
3. Post restoration.

Pre restoration

Before restoring we need to perform certain validations like what type of backup file it is, on which version it has taken, whether existing backup will be supportable on new server (or) not.

To check restoration

Restore verify only from disk = 'E:\1 backup\sqltest.bak'

Restore verify allows sql server to validate database backup without actually performing the restore operation.

It gives output whether backup file is valid (or) not.

If the value is 1, it means backup is valid we can go ahead and restore this backup.

Restore file list only

File list only gives details of backup files like how many files it is going to create, what is default location of backup, size of backup file, logical file name.

Using these details if we want to change path according to new server and drive we can modify.

Restore Filelist only from disk = 'E:\1 backup\sqltest.bak'

Actual restoration

Restoration is the process, From the backup files we are copying the data to another server. We need to specify backup file name and location and as we are copying from data storage location specify the command "From disk". Full backup contains mdf file and Ldf active portion, while restoring the backup it follows 3 steps:

1. Schema initialization, database structure will be created
2. Performs data copy into structure from backup.
3. Performs recovery process on the database files.

Restoration command

```
Restore database sqltest2014 from disk =  
'E:\backups\sqltest-full.bak'
```

For restoring log backups there is no separate restore command. We use same command for all backups. SQL Server identifies with same syntax, what type of backup it is and restores data accordingly.

Post restoration

After restoring we have verify.

1. Size of the database
2. Who is the owner
3. What is compatibility model

We are comparing these details from source and destination. With what ever login we are restoring the database it becomes owner of the new database.

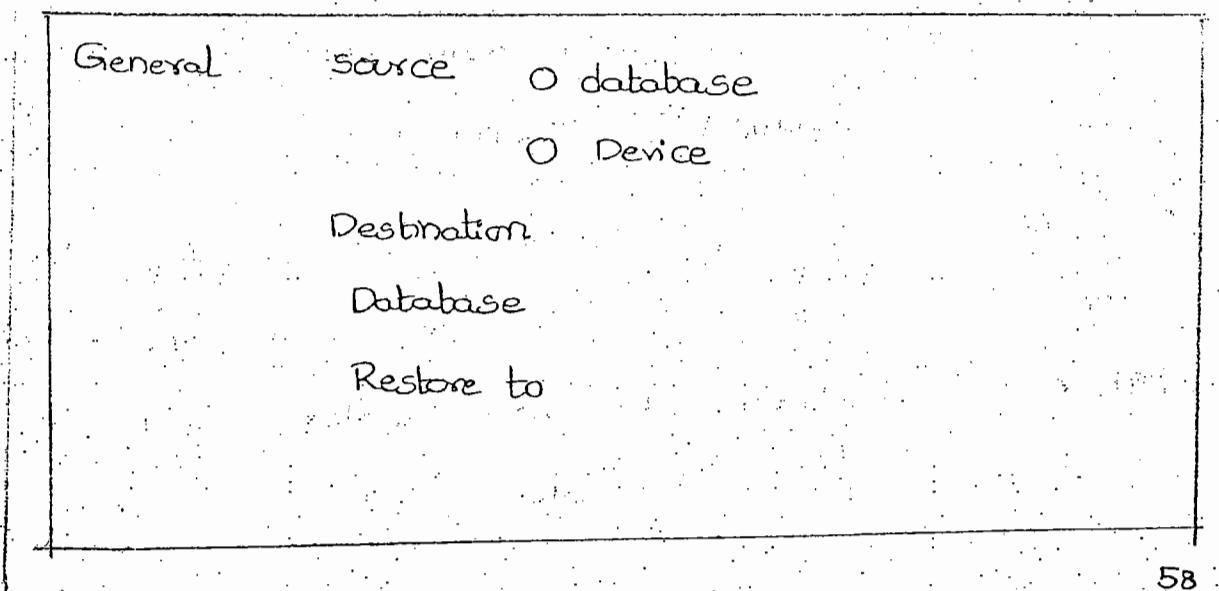
We have to change the owner after restoration by using command

SP- change dbowner 'sa'

Restoring backups using SSMS (GUI)

Right click on the database → Go to Tasks

→ Select Restore → Select database



Source database → We can select the database list for copying the database

Source device → Provide the backup file path

Destination database → What should be the name of name of database that is going to be created. In case database already exist if we want overwrite select the database from dropdown.

Restore to → To stop restoration till previous time we can specify time

- Restore options
- Overwrite existing database
 - Preserve replication setting
 - Prompt before each backup
 - Restrict access to restored database

Overwrite, it replaces old data with new backup data

Preserve replication, If we enable this option along with user tables some replication objects will automatically come with backup.

Prompt before backup, When one backup completes we get notification using this option.

Restrict access, To provide high security while restoring we enable this.

Restoration options

With replace

With (Move)

With stop-at

With replace

There is a database on the server, If we want replace the data on existing database we use this option. It will delete old copy of database and copies a fresh copy of data on existing database.

Through command

Restore database database name from disk = 'Path' With
replace;

Through GUI.

Right click on database → Tasks → Select restore under database → Go to options

Select overwrite the existing database

This overwrite the old data with new backup data.

When ever we want to replace old data we can use replace option and when ever existing Mdf and Ldf not working properly due to some issues here also

We can use replace old one with new backup data

With Move

Backup copies the filename and properties with path, while restoring we can change the paths according to new server drives using with move option.

Situations

1. On source server drive letters are different and on destination server drive letters are different, in this we change with move option and specifies drives according to new server.
2. On destination, if we don't have space on default backup path then we use move option.
3. As per new database server requirement.

Using Command

Restore database database name from disk = 'path'

With move 'logical name' to 'path'

move 'logical name - log' to path

Whenever we use with move while restoring, first need to collect information of logical name and number of files in restoration using filelistonly command.

From GUI

Change default path in options, provide new paths.

With stopat

This we call as point-in-time restoration.

Whenever we want to restore the backup at particular time only. It can be used only for log backup's.

Situations

1. Whenever any client / user ask to restore data at particular time we use this option.
2. Whenever any data deleted accidentally, we have to restore backup before 1 minute to recover the deleted data from existing backup

Steps in restoring

1. Restore recent full and differential with no recovery
2. Restore all T-log backup in same sequential order
3. Restore last T-log with 'stop-at' clause

Using LSN and timestamp we are specifying the restoration to stop before deletion (or) specific date and time. It validates time stamp whenever it reaches stop-at time it stops restoration, the remaining part of the backup will be ignored.

Through command

Restore log dbname from disk = 'Path' with stop-at 'Timestamp'

Restoration situations

Situation 1

If we forget to keep recovery option with last restoration what will happen.

Database will never come online, if we forget to keep last log back recovery option. We have to use Recovery command to bring database online.

Restore database SQLtest2012 with recovery.

Situation 2

If we lost recent differential backup. How will you recover.

We can restore the data with full and followed transaction log backups. Differential will not impact LSN chain.

Number of restorations will be minimize using differential.

Situation 3

Errors while restoring.

- When ever insufficient disk space restoration fails.
- Operating system 302 error when drive (os) path not exist.
- Access denied error when we do not have sufficient permissions while restoration.
- Restoration terminated abnormally with errors.
 'Unable to get an exclusive access on the database'
 As lock is there it not allow restoration.

States and steps of Restoration

1. While restoring it perform copying the structure of backup's (mdf and ldf file structure)
2. Dataload, copies data into backup file structure.
3. Performs recovery ① Roll forward ② Roll back

Roll forward → Full backup contains Active ldf portion
While restore it copies the data, after completion
While starting of database sql server performs
recovery process it verifies whether any pending
transactions are there on the server, those will make
a move to Mdf.

Roll back → uncommitted data will flush as part
of recovery. Once this recovery process complete
then only database will start.

No-recovery

Whenever data load is not complete we will restore
backup with No-recovery option. When there is no
backup left for restoration we use recovery option

Restoring Enhancements in sqlserver 2012

In sqlserver 2012 Microsoft introduced database restore enhancements

1. Point - in - time
2. Page restore

Point in time restore has now a visual time line that allows us to select the target time and we can perform restore. Earlier we can do this only thru script.

SSMS → Right click on database - select Restore database

Select 'sqlserver 2012' backups available on servers

Now all the backups of sqlserver 2012db will display in list. There is new button called 'TIMELINE' click on it to open timeline interface.

choose a backup and select option specific date and time.

With New Timeline feature we can scroll to restore time that we want. We can which type of backup it is also. once selected correct time click OK. It generates restore till requested time.

Backup Timeline

Restore to : Last backup taken Specific date and time

Date Time

Time Line Interval 12:00 13:00 0:00 6:00 12:00 16:00

Page restore

We have Page restore option in SQL 2005 and SQL 2008

also ~~but it has~~ now user interface. It allows
us to check database for corrupt pages and restore
them from a good backup file.

Right click on database → select Tasks → Restore
→ Page

In Restore Page window, the database and the
Pages grid will show the damaged pages.
by DBCC 'check database Pages'.

The repair Pages grid displays records that appear
in suspect - Pages table in msdb database.

We can execute DBCC CHECKDB with physical - only
command against database to populate the grid.
by clicking the Add button.

SECURITY

Security is the most important to.

We need control the access to sqlserver as well as provide security to the data.

These are mainly 3 levels of security.

1. Server Level

2. Database level

3. Object Level

Server Level is the topmost Level in connecting to Sql server.

Database Level, once connected to the server, we need to provide security to user at database level.

Object Level, once connected to database, we need to provide security on database object like tables stored procedures etc.

For providing security we have 3 process

1. Authentication

2. Authorization

3. Encryption

Authentication, Validating credentials on server

Authorization, Permissions to perform their operations.

Authentication

Validating Credentials (User name and password) on server called Authentication. For any User who are connecting to server will have login name and password. We have 2 types of Logins

1. Windows login
2. SQL Server Login

Domain user, Database Administrators, other admins, developers will have Windows login for connect to Server. These users use same account for connecting to windows and SQL Server. Windows login store in Windows Component called Active Directory. These logins provides high security.

SQL login, For the users who are connecting externally from frontend we are providing SQL login. They first connect to windows and then connect to SQL Server. These logins stores syslogins table of Master database.

We need to provide security to SQL Server. We have to validate both Windows and SQL logins. This process of checking login name and password is Authentication. There are 2 types of Authentication in SQL Server.

1. Windows mode Authentication

2. Mixed mode Authentication

Windows mode

It allows only window login users in this mode. It enforces only windows user to connect and access the databases. When we use highly restricted database we go with windows login. We do not provide access to external users if the data is highly restricted. This mode provides best security. It validates logins in Active directory.

Mixed mode

This mode allows both windows and sql logins. Whenever any login tries to establish connection first it validate user name and password in Active directory. If login exists, establishes connection otherwise validation goes to syslogins of Master database and verifies there. If login exists it will establish a connection.

Whenever any request comes first we need to verify the login Active directory. If it is not exist we need to create (or) ask Active directory admins to create.

Command for windows

Create login [Domain\loginname] from Windows 69

Authentication at database level

Login Account is a Key to Main door means to connect to server. It's a Server level access.

We require another account called user account to access the databases inside the server. This is a database level access. We need to grant access for these user account on each database separately.

User mapping

Every login account will be mapped with the user Account

Every login will have a security id (sid), it's a unique security number inside sql server to avoid duplication of logins using the sid of login account

We will map to sid of user account. Now it establishes relationship between login account and User account, this is user mapping. One login account will mapped to one user account only. We call this as one to one mapping.

Once the user mapping is created between login and user account, sql server allows to connect to server and then to access the databases inside the sqlserver.

Creating Windows Login

Create login Loginname from windows

Creating SQL Server Login

For creating any login, it requires 4 properties

1. Login name
2. Password
3. default database
4. default language (optional)

Using T-SQL

SP- add login 'Loginname' 'password' 'default database'

Ex:- SP- add login 'SQL2012Admin' 'sql@12345' 'Master'

Using GUI

Go to security folder under Management studio → Go to login → Right click on login → select New login

General	<input type="radio"/> Windows login <input checked="" type="radio"/> SQL Authentication
	Name
	Password
	<input type="checkbox"/> enforce password policy
	<input checked="" type="checkbox"/> enforce password expiration
	<input type="checkbox"/> User must change password at next login
	Default database
	Default language

Domain users will set these password policies. These policies will be enabled at window server level. We are using same policies for sql login also. Once we provide require properties to login we can enable password policy. What is minimum password length and complexity.

- Enforce password policy

We are using password policies which created at domain level

- Enforce password expiration

When this login is to be expire. It expires after specified days

- User must change at Next login

Domain users create login and provides password

We need to change the password in logging first time.

When user login next time sql server prompt a message to change the password.

Once all these details provided login will be created.

Creating user account.

Authorization

Once user connected to server, they should have permissions to perform / operating certain tasks. As a DBA we need to restrict the permissions by granting the roles, roles are set of privileges for connection server and to access the database.

We have 2 types of roles

1. Server role.
2. Database role.

Server role to perform server side operations. We have 8 server roles and by default it will be public role.

Sys admin : Top most privilege on the server, capable of doing any thing on the server. DBA's will have this role.

Server admin

Allows users to manage configurations on the server. Capable of start and stop services from configuration manager, server settings, memory configurations and shutdown of sql servers.

Setup admin

Setup admin capable of creating linked server. He can run any setup utilities like replication, mirroring.

Security Admin

Capable of creating new logins, delete logins and reset Passwords. He has ability to grant, revoke and deny Permissions.

Process Admin

monitors the process running under Task bar, capable of killing problematic process

Disk Admin

Manages storage. If any new drive need to add, he has permission to add new drive to the Server.

BULK Admin

Capable to performing bulk inserts and updates.

Db Creator

This role is capable of creating databases, he can alter and drop. Taking backups and restore.

Public : The default role that server role consists of.

Using TSQL

sp-addsrvrole member 'loginname' 'privilege'

Ex: sp-addsrvrole member 'Empadmin' 'sysadmin'

Using GUI

Right click on login → go to properties → server roles - select role

Database roles

Db_owner

Db_owner is capable of doing anything on particular database. Highest privilege at database level.

Db-Security admin

He is responsible for creating/deleting/modifying roles to users and manages permissions.

Db-datareader

He can run select queries on all tables, only read activities.

Db-datawriter

He can perform insert/update/delete data in the database.

Db-deny datareader

To restrict the read access on particular confidential data even though user has server level (or) high privileges at database level we can prevent that user in accessing particular database using deny role.

Db-deny data writer

He cannot able to write data on restricted databases.

Db-DDLadmin

He can able to create/Alter/drop the tables and other objects.

Db-access admin

He can not able to create (or) modify any user role. He can grant only roles and remove access to other users.

Backup Operator

He can able to take backup of databases.

public : This is default role on database level.

Using T-SQL command

SP-addrole member 'username' 'rolename'

Ex: SP-addrole member 'Testadmin' 'DB-Owner'

Using GUI

Right click on Login → Go to properties → Go to user mapping → select database and grant database role.

Along with these fixed database roles, in SQL Server 2012 we have flexible roles.

Flexible roles are the roles to create group of users in terms of functionality.

SQL Server 2012 introduces contained databases.

There are 2 types of authentications in this

1. Contained databases users with passwords, stores all the passwords within contained database
2. Contained databases Windows is similar to windows login.

Object level

We need to identify 3 things in providing object level access.

1. Principals are login accounts to whom we are granting permissions. Login accounts and user account comes under Princip
2. Privileges, how we granting permissions to access objects.
3. Securables, on which object we are granting Tables, Stored Procedures and views

Through T-SQL

Grant/revoke permissions on securables to princ

Ex Grant insert on dbo.Sqltest to empadmin

Through users

Go to users under database → go to properties → go to securables → select objects from search
→ object types provides list of objects Tables, views etc.
Select them → Provide permissions and click ok.

Encryption

Using encryption we can protect the confidential data while transferring data through networks. It restricts hackers to access data.

1. Encryption using certificate.
2. Encryption using Asymmetric keys.
3. Encryption Using Symmetric keys.

Certificate Encryption is used to identify users, devices and organization. Certification Authorities generate these certificates to provide high security. SQL Service will use these certificates and encrypts data.

Asymmetric keys uses a public key that encrypts data before sending and private decrypts that message after receiving. Here we have 2 keys in encryption.

Symmetric key uses same key to encrypt and decrypt the data.

Transparent Data Encryption (TDE)

TDE is a SQL Server 2012 enhanced feature that allows encryption of data in better way.

AUTOMATION

Automation refers to scheduling the regular maintenance tasks. Whenever schedule time comes automatically that task will be executed. DBA regular activities can be automated. We can get the status of the task in the form of alert and helpful while generating reports.

Automation can be done in 2 ways

1. Jobs
2. Maintenance plans

Jobs

SQL Server Agent jobs useful to perform daily scheduled maintenance tasks. A job consists of series of steps that work together to perform a task.

- All these jobs falls under Database maintenance category.
- We can start, stop, enable, disable job as per our requirement.
- Easy to create and maintaining jobs. Whenever any issue happen we can trouble shoot easily.
- Scheduling also easy with jobs.

How to Create a job

Right click on the job folder under sql server Agent

→ Select New job from options menu

General

Name : Full back up

Owner : SA

Category : Database Maintenance

Description : Perform full backup at 6AM

Enabled

General helps DBA's to provide identification of jobs.

Name → We need to provide name for the job

Owner → Always job owner is to be sysadmin (SA)

Category → Under which category this job falls, Mostly all DBA tasks falls under database maintenance.

Description → Provide some description about the job

Enabled → Job will be created now, we can enable later using this option.

Steps

Steps are nothing but a task under job.

Select New button.

Job Step General	Stepname	Backup
	Type	T-SQL
	Runas	
	Database	
	Command	Code
	Open select ALL Copy Paste Parse	

Stepname → Provides details of step of job.

Type → SQL Server uses T-SQL language by default

Runas → only that privileged persons can able to run this job, if we specify.

Database → By default it points to Master, we can change as per requirement.

Command → T-SQL command to perform the task

Parse → We can verify syntax errors of the T-SQL command using this option.

Advanced	on success
	Retry attempts [2] Retry interval [5] minutes
	on failure
	out put file E:\jobs\output.txt
	<input type="checkbox"/> Append out to Existing file
	<input type="checkbox"/> Include step output in history

On success → Provide next steps once the first step completes means it has to go to next step (or) it has to quit the job.

Retry attempts → No of retry attempts need to specify. If a job fails it will attempt try for specified times and report to user if fails again.

Retry interval → Wait for specified time and attempt next try.

on failure → on failure where this job have to go whether it has to start from beginning (or) quit the job reporting failure to user.

Output file → We configure output file to get the information about the job. If it fails with what error it fails we will get from this.

Append output → overwrites the existing job information

Log to file → If we want job running status data in the form of table

Output in history → To view history of job execution

We can add edit, delete steps, move the order of steps using insert, edit and Delete buttons

schedule	Name
Schedule type	<input checked="" type="checkbox"/> enabled
One time occurrence	
Occurs	
Recur every	
Occurs every	
Start date	

Name → Name of schedule we need to provide here.

Schedule type → Whether it is one time job (or) daily job we need to provide type.

enabled → Scheduled it to be enable now (or) later.

Occurs → Whether it is daily (or) weekly (or) monthly.

Recur every → It will re-run the steps on specified day.

Occurs every → We need to specify the frequency of executing this job means every 1 hour (or) 1 hours as per our requirement.

Summary of job will be displayed at the end.

Once all the details provided click OK, it will create a job with specified name.

Post job steps

1. Once the job is created we can start, stop, enable, disable and delete job by right clicking on the job.
2. We can edit the job if we want.
3. We can script out the job and copy the script for creating same job on other servers during migration.

Right click on job → select script job as → open job in New query window → copy the script of job.

Place on different server and execute

Trouble shooting the job failures

1. Output file → open out file which we configured while creating jobs under Advanced steps of job. It records all the details of job for troubleshoot.
2. View history → Right click on job goto view history. Provides details of job execution. If fails with what error it failed we will get all details here.
3. Verify in Msdb tables like sysjobs, sysjobhistory
4. Go through sqlserver error logs for failure notification messages.

Migration

Migration means moving one (or) set of databases from one server to another server.

Scenarios for migration.

Situation 1: As a part of Launch, Deployment, go live

→ Migrating databases from development to test and Test to production. This we call as Launch.

→ Whenever new builds released to production as a part of enhancement we call as deployment.

→ When user can access the application online we call as Go-live

In all these 3 types we are migrating databases from server to other server

Situation 2: Database refreshes

Moving Production server database copy to develop environment for building next next module. They test with production data. Using prod data they design new stored procedures as part of next requirement. We move Prod to Test also. Here Testers will perform ad hoc testing on data for tuning query.

Regularly we are providing the production database backup for building next enhancements and testing of some stored procedures and queries.

Situation 3 : Moving from old hardware to New hardware

Whenever we are buying system it comes with a warranty. Once it reaches the EOL [Guarantee and] we will check with vendor who sells system. If vendor guarantee further we can continue otherwise we have to move all databases from old servers to new servers.

Situation 4 : Migration between data centers.

Data center is a place where we are physically keeping servers at particular location. We can have multiple datacenters. We are maintaining production datacenters and DR datacenters. As a part of High availability configuration we are moving all server databases from one datacenter to another datacenter.

- As a part of making DR server, we move all databases from one datacenter to new data center.
- Whenever we are shutting down old datacenter we move data to new data centers.

Situation 5 : Consolidation.

Instead of storing Non critical databases on more servers, we move them to one big server. here monitoring will be effective but performance may effect.

Steps in Migration

1. Transfer Logins

Before moving the databases, script out the logins on source server and keep them aside. There is a stored procedure `sp-help-revlogin`, we have execute this stored procedure in the server. First it creates the stored procedure under master database. This will not come as part of SQL installation. Once the stored procedure created we have to run the stored procedure second time it will script out all logins of source server including password in encrypted format.

We can copy all logins (or) few logins according to our requirement. Copy all the logins to destination server and execute them on server. Same login and password will be created.

Second way, Right click on login and script as copy them on other server. This process will not copy password.

2. Transfer jobs

Right click on job → script job as → create to New window, copy the script and run on other server.

Whenever we want to transfer all jobs we need create a package and transfer that package.

3. Transfer the packages.

We have to connect to Integration services → Go to Msdb database → under stored packages we will have SSIS Packages. → Right click on Packages → on the Package Wizard select Export package
Provide destination server location.

Now connect to destination server →
Go to Integration services → Go to stored packages → under Msdb right click on Packages and select Import package, Package will be created on server.

4. Transfer the database.

(i) Back up and restore

Take backup on source server and copy them to destination server.

Right click on database in destination server → select Restore and provide backup details and click OK
Backup and restore is online, source will not be distributed here. Roll back is easy.

(ii) Attach and Detach

Right click on database of source server → Go to task → select detach database, database will go offline
Copy files from source server and paste them
Go to destination server → select attach - Provide path details

5. check new database properties

Verify the database size from database properties

From options verify - Recovery Model, db owner, compatibility model and collation settings

Change the db owner using sp_change_dbowner 'SA'.

Find and fix orphaned users

Whenever taking full backup and copy & restore on destination server. All User account will copy to destination server. User account without corresponding login account is called orphaned users. As the login will present in Active directory (or) syslogins of Master db. Some times synchronization will miss even after transfer logins also.

To find orphaned users.

SP-change - User - login 'report'

To fix

SP-change - users - login 'update-one' 'username' 'loginname'

Update one will establish synchronization between Username and loginname.

Moving system databases.

start sqlserver in single user mode, Go to start up parameters and type -m

From command prompt

C:\> Net start mssql

Once SQL start in single user mode

Restore database master from disk = 'Path' with replace.

Whenever we restore Master database we can do it only in single user mode. Restore Model and Msdb after removing (-m)

From configuration manager → Right click on Services → Go to Advanced → click on startup parameters
Go to end type -m and apply then restart services.

Migrate tables

Right click on database → go to Task → Export data

It will open import and export wizard.

Specify Data source and server name, database.

→ select copy data.

→ Select tables which we want to migrate.

We can do modifications to size, datatypes here.

Run immediately.

Migrate stored procedures

Right click on database → Go to Tasks → generate Scripts → select a database and select objects, stored procedures → Select All → Script to new window (script to file)
Copy the script to other server and execute these scripts on them.

Copy/Migrate to Excel

Connect to SQL Management Studio → under Management folder → Go to Export option by rightclicking → SQL Import and Export Wizard will appear

Choose data source → SQL Server Native Client 11.0

Server Name → Provide Server Name

Select Authentication → Windows SQL Authentication

Database → Select database

Click Next (Go to Next page)

* Choose destination → provide Microsoft Excel

(Server name → Provide destination server name)

File path (Authentication) → Provide Excel file path

(Database → Provide destination db)] q1

choose destination → select Microsoft Excel

Excel file path → provide location to store Excel
and provide Name to file

First row has column names

Specify Table copy → select Copy data from
one or more tables or views

If we want we can edit Mapping.

Click - Finish.

Same way we can import data from any other
source to SQL Server

Excel to SQL Server

Oracle to SQL Server

Flat file to SQL Server

SQL to SQL Server

SSMS → Right click on Management → Go to Import Data

DataSource select Excel - Provide paths of Excel file

Choose destination → Select Server Name

Provide database Name →

Provide New table Name to data

Save and Run Package → Finish.

Upgradation

Upgrading means migrating sql server old version to sql server 2012.

Before upgradation we need to run Upgrade Advisor Tool.

Microsoft SQL Server 2012 Upgrade Advisor.

under Installation menu → In planning → click Upgrade Advisor.

SQL Server 2012 Upgrade Advisor analyzes SQL Server 2008 R2,

(or) other lower versions SQL Server instances and gives results that may impact upgrade process.

Steps in installing Upgrade Advisor.

1. Open Installation → Go to planning → select Upgrade Advisor → Wizard will be open → click Next.
2. Enter Server name and select the components to analyze.
3. Select Instance name and provide credentials.
4. Select the databases; If you have SSIS packages provide path of files.
5. Select run to begin analysis.
6. Once Analysis complete, warnings will be displayed.

There are 2 types of Upgradation:

1. Inplace upgrade
2. Side-by side upgrade.

Inplace upgrade

Source server will become destination server after upgrading. It is going to install SQL Server 2012 Components on old version. All the lower version support files will be converted to SQL Server 2012 supportable. This type of Upgradation is Inplace upgrade.

Advantages

1. Fast, Easy and Automated process.
2. No additional hardware required.
3. Instance name will be same as earlier.

Disadvantages

1. SQL Server will be in offline state during upgrade hence we have downtime.
2. Roll back process is very complex.
3. Source server may be effected.

Inplace upgrade process

From Installation → Select Upgrade from SQL Server 2008 R2 and continue remaining installation process.

Side - by - side

Side - by - side upgrade consists of installing sqlserver 2012 and moving old server databases to new server just like migration process.

Advantages

1. Upgrade process is online without effecting business.
2. Minimal downtime required in redirecting the user requests from SQL 2008 R2 to SQLserver 2012.
3. Selected databases can migrate.
4. Full control on upgradation.
5. Roll back will be easy as original server still available.

Disadvantage

1. Manual process.

Side - by - side process

1. Follow the same process which performed in migration like script out logins, jobs and take backup and keep.
2. Install sqlServer 2012.
3. Copy the database backups and restore.
4. Create logins and jobs by running scripts.
5. Find and fix orphaned users.
6. Verify database connectivity and functionality.

Post-upgrade steps

1. change compatibility level
2. check integrity of objects
3. set page verification using CHECKSUM.
4. update statistics.

After upgrade change the compatibility level of database

ALTER database databasename set compatibility_level = 110

Check integrity

Check integrity of objects in a database using DBCC
CHECKDB

DBCC CHECKDB (Database name)

Page verification

Checksum verifies computation of pages and results stored in page header

ALTER database databasename set page_verify
CHECKSUM WITH NO_WAIT

Updating statistics

The statistics need to update after upgrade to get full features of enhancements in sqlserver 2012.

sp_msforeachtable 'UPDATE STATISTICS ? WITH FULLSCAN'

LOGSHIPPING

Logshipping :-

Shipping of Transaction log files from primary server database to second server database to get both databases in continuous synchronization. It is a continuous process in the form of batches.

Logshipping requires 3 servers

1. Primary : Primary server contains source database for sending the transaction logs.
2. Secondary : Secondary server contains destination database receiving the transaction logs.
3. Monitor : Monitor server tracks information related to log shipping jobs and sends alerts in case of failure.

Logshipping mainly depend on 3 jobs

1. Backup job

For Backing up transaction logs on primary server database.
Always run on primary server.

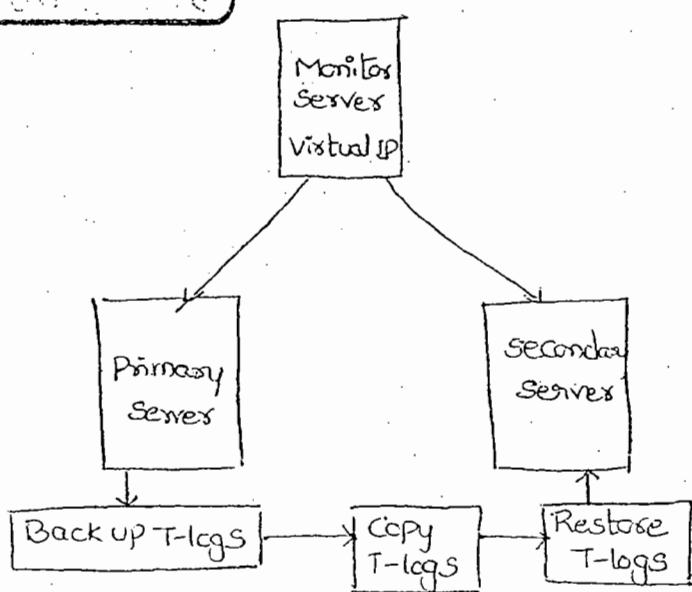
2. Copy job

copy job runs from secondary server, copies transaction log backup from primary server to secondary server.

3. Restore job

Restore run on secondary server to restore backups.

Logshipping Architecture



Logshipping steps

1. Before logshipping configuration, Take full backup of Primary database and restore it on secondary server
This is initial synchronization.
2. Create Backup job on primary server to backup the transaction logs of primary database.
3. Create copy job on secondary server to copy the transaction logs.
4. Create restore job on secondary server to restore the transaction logs.
5. Create an Alert job on monitor server to indicate a failure if the databases are out of synchronization beyond the configuration threshold.

Failover steps

Planned failover

- 1. situations 1. upgrade from 2008 R2 to sqlserver 2012
- 2. Hardward failure ,

In these situations. Keep primary server in offline and make secondary server database online.

1. Disable all logshipping jobs on both servers.
2. Make sure copy all backups copied to secondary server
3. Restore all T-log backup in same order
4. change connection to secondary server and start jobs.

unplanned / Disaster

Need to identify crash time , when last backup, copy and restoration happened on server using 'dbo.logshipping-monitor' tables of MSDB databases.

1. Disable all logshipping jobs
2. Perform Tail log backup if log files available
3. Manually copy last backup's from Network share to secondary server.
4. Restore all T-log backups in same sequence.
5. Now secondary database will come online
6. Sync all logins, jobs then find orphaned user and fix them.

Advantages

1. Only logshipping allows to keep secondary database in stand by mode.
2. We can maintain multiple secondary databases.
3. No additional hardware or software required.

Disadvantages

1. More down possible, we cannot bring secondary server immediately.
2. Data loss possible, as jobs run every 15 minutes.
3. We have to perform manual failover.
4. Need to perform several steps to bring secondary server online which we call from stand by.

Components of Logshipping

For Configuring logshipping We require 2 servers and 3 jobs and we need Backup destination, There are 2 types

1. Local share
2. Remote (a) Network share.

In local share, back up copies into Primary server share

When ever Primary goes down dataloss is more.

In Remote share, Backup copies to Network share. Copy job of Secondary Server goes to this share and copies back up to Local share of secondary server. We have to grant proper permissions to SQL Agent services for this Network share folder on both servers.

Pre-requisites

1. Make sure recovery model should be Full.
2. Maintain same hardware between both servers to avoid performance issue.
3. Service Account running on both servers must have access to Network share where log backups store.
4. Keep Secondary server always in No recovery.
5. Make sure Number of drives and Name of drives should be same on both servers.

Configuring Logshipping

1. Select database → Go to properties → select options
From Recovery model drop-down choose Full recovery model

By using T-SQL

use Master

Alter database databaseName set RECOVERY FULL;

2. From Properties → select Transaction Log shipping

Enable this as a primary database in logshipping

Click Backup setting button

3. Transaction log backup settings

Provide Network path to backup folder and

Local path if folder is local to primary server.

4. Using schedule option we can change Time interval

By default it will be 15 minutes (Latency).

5. Delete files older than (Retention period) hours

Alert if no backup occurs within hours

6. Keep Compression as default.

Based on retention period old backup files will be deleted

Whenever there is no backup for particular period we

need to get Alert.

Backup setting completed, secondary server settings enabled.

102

Transaction log back settings

Network path to backup folder → \\Win 2k13\Primary backuplog

Local path to folder → \\C:\Primary\backuplog

Delete files older than → hours

Alert if no backup occurs with in hours

Backup job name → LSPbackup- databaseName

Secondary Database settings

Select secondary server name :

Initialize secondary database | copy files | Restore T-log

Initialize secondary database

We have to create database on Secondary server, if not present.

Select one option from the 3 options.

O Yes, generate a full backup of Primary and restore

Create secondary database if not available by using above option

O Use existing backup of Primary into secondary

In case backup already exist we use this option (It is not recommended, LSN mismatch may happen).

O Secondary database is already initialized

Restore options

In this we specify location of Mdf and Ldf files.

Copy files

Destination folder for copied files: \\win2ks13\secondary

Delete copied after (retention period): 24 hours

Provide name to copy job

Schedule setting → Specify setting

Restore Transaction Log

No recovery mode will not allow users to access secondary db.
Stand by mode, allows users to run select commands till next restoration.

Disconnect users when restoring, kills all user connections.

Restore T-log

No recovery Stand by Mode

Disconnect users in db when restoring

Delay restoring backup at least

Alert if no restore occurs within

No Recovery → To keep database in restoring mode

Stand by → We can execute select commands

Disconnect users → It kills user who connect to server while restoring

Delay restoring → If we want wantedly delaying restoration.

Logshipping configuration completed

Failover steps in Logshipping

1. Disable all logshipping jobs on primary server.
2. Check whether we can take a tail log backup or not.
If log files available then only we can perform tail log backup.
3. Manually copy last backup and Tailog backup from Network share to secondary server.
4. Restore all available log backups with no recovery and last log backup with recovery.
5. Now secondary database will become online.
6. Sync all logins, jobs
7. find orphaned users and fix it.

Advantages and Disadvantages

Advantages

1. Whenever we have requirement, we can keep secondary database in stand by mode.
2. We can get multiple secondaries.

Disadvantages

1. More downtime and data loss possible.
2. Manual fail over.
3. In logshipping database as warm stand by, we have to do few more steps to bring server online.

TUF file in Logshipping

TUF file stands for Transaction undo file (TUF)

Tuf file contains the modifications that were not committed on primary database when transaction log backup was in progress and when log was restoring on secondary database. When next transaction restoring in secondary server, sql server uses the Tuf file and starts restoring incomplete transactions.

Monitoring Logshipping

1. Rightclick on sqlservername in ssms → Go to reports
→ Select standard reports → Transaction logshipping status.
2. We can query msdb tables
Select * from msdb..log_shipping_monitor_error_detail
3. We can query sysjobhistory
Select * from msdb..sysjobhistory where message like '% operating system errors'

DATABASE MIRRORING

DB Mirroring, maintaining two synchronized copies of databases is called mirroring. Secondary database will be in No recovery always. Only one copy of database is available to users at any time.

Mirroring implemented by combination of other 3 High availability concept.

- (i) Initial synchronization of databases. taking full backup and restore on secondary server taken from logshipping.
- (ii) From replication, Transferring log buffer (info about transactions) from primary to secondary server and commits these and sends acknowledgement to primary. This taken from Transaction replication.
- (iii) Automatic failover concept taken from clustering.

Components in Mirroring

Mirroring requires 2 servers. If we require automatic failover we need 3 servers.

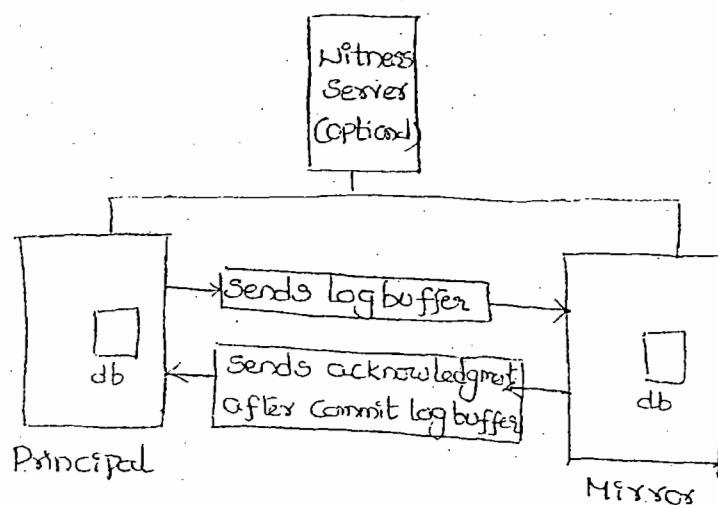
- (a) Principal server : Primary server is a source server contains user transactions.
- (b) Mirror server : Secondary server contains destination database, stores logbuffer that comes from principal.
- (c) Witness server : optional server that monitors Principal and Mirror servers.

Endpoint : End point is a Object to a network protocol

Establish connection and communication between

Principal and Mirror across the network.

Mirroring uses TCP/IP protocol and 5022 port number for network transfer between nodes. We call opposite server as partner.



Steps

1. Write the data to the Transaction log of Principal server.
2. Sends transactions (logbuffer) to mirror server.
3. Write the data to Transaction log of Mirror and Commit the data.
4. Sends acknowledgement to principal.
5. Transactions will commit in Principal server.

How Mirroring Works

Take full and log backup on Principal server and restore on mirror server. Witness server role is to check heartbeat. Any transaction comes to Principal write information in log buffer (log records) and sends to mirror server once transactions commit in mirror database sends acknowledgement to principal server, Now the transactions will commit in principal server.

Entire sending Log buffer and receiving acknowledgement done by Network communication mechanism, for this transfer mirroring uses Protocol → TCP/IP and Port no : 5022 , it configures End point which establish connection between 2 servers The end point name itself is Mirroring .

Whenever we restore data in mirror we keep database in No recovery state , so that mirror always to be ready to accept transactions from Principal. Witness Server continuously pings mirror and principal , At any time if Principal goes down , it brings mirror online to continue the business to users . Quorum is a communication between the servers check whether Principal is up or not . If it is down quorum sends communication to mirror and brings mirror online .

Types of Mirroring

1. Synchronous Mirroring.
2. Asynchronous Mirroring.

Synchronous, Principal server was forced to wait until it receives acknowledgement from mirror server. After receiving acknowledgement then only it commits the transaction fully in Principal.

Users have to wait more time until it commits the data on both servers. Performance is slow but the data will be highly synchronized.

Asynchronous

For High performance in mirroring, once transaction comes to principal server it copies logbuffer to mirror server and commit transaction immediately in Principal Server it will not wait for acknowledgement from mirror. Here we get high performance but synchronization of data is less.

Operating modes of Mirroring

In synchronous we have 2 modes

- ① High availability → whenever we require automatic failover using witness server
- ② High safety → whenever there is no need of automatic failover we can configure.

In Asynchronous we have 1 mode

- ③ High performance → Transaction sends to mirror and commits in Principal without waiting for the acknowledgement. Performance is high but data will not be in completely sync. mostly used mode.

When ever we Configure DB Mirroring by default it goes to High Safety mode. Later we can change to other operating mode.

Configuring Mirroring

Connect to principal server → select database → Go to properties → select mirroring.

Configure security

Principal

TCP: \\ Win2K12 : 5022

Mirror

TCP: \\ WinMIRROR2K12 : 5022

Witness

TCP: \\ WinMIR2K12 : 5022

Operating mode

○ High performance (Asynchronous)

○ High safety with out automatic failover

○ High safety with automatic failover (Synchronous)

Configure security

Go to service Accounts if Principal and Mirror are running with different service accounts mention them other no need to enter.

Once we provide Principal, Mirror and witness servers FQDN Paths click on start mirroring, it configures end points and then start mirroring.

Pause → To suspend mirroring in middle.

Resume → To start mirror from pause

Remove mirroring → It breaks mirroring between servers.

Failover → Principal becomes mirror, vice versa.

Types of failovers

1. Automatic failover

Witness will detect heartbeat failure, we use it in High availability mode.

2. Manual failover.

Using T-SQL script

Alter database databasename set partner failover

Using Gui → Right click on database → Go to properties

Go to Mirroring → From Mirroring Wizard → click Failover

3. Forced failover

Forcefully bringing mirror online, cancel recovery process

Alter database databasename set partner failover with force=allow

- dataloss

Mirroring Situations

Situation 1

When Principal server down what will happen in Mirroring.

When ever Principal server down, witness server identifies heart beat failure Quorum Establishes communication between devices and brings mirror server online.

Situation 2

What happens if Mirror server down.

When ever mirror server down, quorum establishes communication and ask principal server not to send transaction. It will be in UNSENT LOG position in mirroring. The changes pending for commit in mirror is called unsent log position that only committed in Principal.

As soon as Mirror server up it establish connection and reads all data in unsent log position and try to sync the data between servers. It will copy all transactions to Mirror server.

Situation 3

What happens if witness server is down.

When witness server down we can not able to perform Automatic fail over, mirroring will be running.

Situation 4

In case of Mirror is down for more days.

We need to break mirroring as each and every transaction will record twice in inactive portion of log file and unsent log position. So there is a chance of getting Principal server log file full.

Situation 5

Why can't we keep both servers online.

When we keep mirror server online, If any user performs transactions on mirror LSN will be break and syncup of transaction between principal and mirror may fail.

Situation 6

How to bring mirror server up immediately.

Mostly in production environment when ever issue comes we can break mirror between servers and run restore command with recovery it will bring mirror server immediately online/up to continue business.

REPLICATION

Replication mainly used as a Data migration Technology although it is one of High availability option. The Concept of replication is to migrate data and database objects from one server to other servers.

Why replication is a Powerful migration Technology

i. Replication can be used for Multiple purposes

(a) High availability

We can maintain 2 servers, both databases are in completely sync. Whatever changes done on primary goes to secondary.

(b) Scalability

Load distribution is possible in replication as both primary and secondary servers are available to users. We can split users among the 2 servers.

(c) Migration

After initial synchronization between 2 servers databases replication tracks the changes and these changes migrate from source server to subscriber server instead of entire database.

2. Replication have filter mechanism, while moving data from one server to another server we can use a filter (or) condition like dept = sales (or) Emp salary > 1000. While migrating data replication checks whether data meeting condition (or) not. If meet data will migrate, if not it ignores the data. This is completely lack in other migration technologies like Backup and restore, import and export.

3. Replication can be done on object level, we can select tables, stored procedures, functions as well as along with data.

4. Replication can be configured between heterogeneous databases. We can migrate data from SQL to Non SQL servers like Oracle, DB2, Teradata etc.

5. Replication has scheduling option, we can define schedule with high latency (or) low latency

How Replication Works

Replication requires 3 servers and 3 databases along with it require Articles, Publications and Agents.

Publisher :-

Publisher is a source server containing primary database called publication database here we can select entire database (or) few tables, stored procedures of database.

Subscriber :-

Subscriber is a destination server. Whatever data coming it stores in subscription database.

Distributor :-

Entire replication process carried in distributor server. It establishes connection with publisher identifies data ready for replication copies data to subscriber.

(i) Local distributor

Publisher and distributor on same machine.

(ii) Remote distributor

Publisher and distributor on different machines.

Articles :-

Articles are the database objects that being replicated.

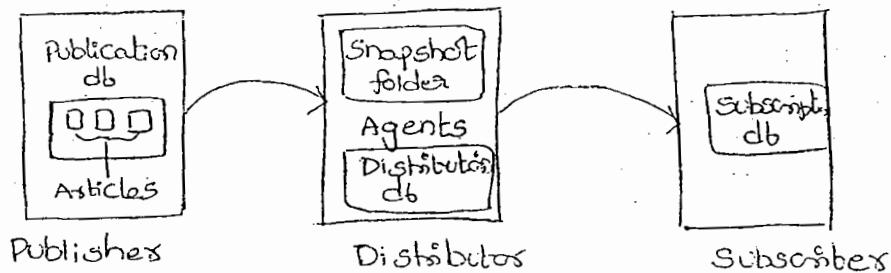
Publication :-

Publications are collection of articles

Agents :-

Agents are executables (jobs) used to perform the replication operations.

Replication process



Replication uses News paper terminology, Publisher (News printing person) have publication database in that Articles means tables, stored procedures will be available for migration.

Distributor (Delivery person) collect the articles from publication database and forward the same data to subscriber using agents and distribution database.

Subscriber (Receiver), receives all the data and keeps in subscription database.

When we configure replication, we need to get (like) details.

1. Replication model

2. Data flow

3. Latency

1. Replication models

- (a) Publisher, subscriber and local distributor
- (b) Publisher, subscriber and remote distributor
- (c) Central publisher and multiple subscribers
- (d) Multiple publishers and central subscriber
- (e) Hybrid (combination of any 2 above models)

2. Dataflow

We need get the details whether flow is one way (or)

Two way.

One way → Data flows from publisher to subscriber.

By way → If both servers in use need to configure replication. By way as the data moves from publisher to subscriber and gets data from subscriber to publisher.

3. Latency.

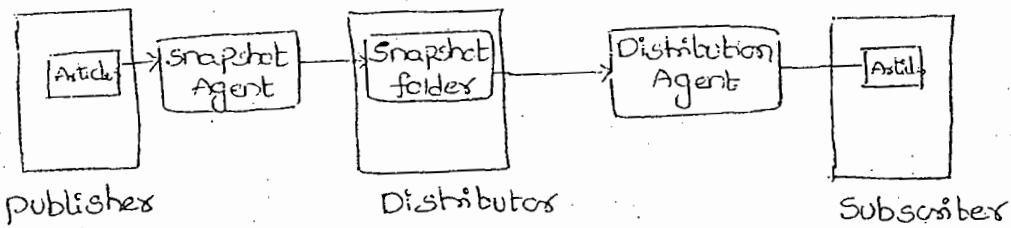
Latency is a time interval, schedule the data movement either continuous (or) hourly (or) weekly.

Types of Replication

1. Snapshot replication
2. Transactional replication
 - (i) Transactional with updateable subscriptions
3. Peer to Peer replication.
4. Merge replication

Snapshot replication

Snapshot is going to make exact copy of an existing database. It is one time replication.



2 agents. Snapshot agent and distribution agent for configuring snapshot replication. Snapshot agent will perform snapshot generation.

Snapshot generation Process

1. Snapshot agent establishes connection between Publisher and Distributor.
2. Snapshot agent identifies what are objects participated in replication process applies schema locks on those objects while copying structure.

3. Snapshot generates script of tables and output will be copied as `tablename.sch` snapshot folder is a storage location

Data will copied with .bcp

Triggers copied with .trg

Indexes copies with .idx

Other constraints copy as .pre

If there are 5 tables participated in replication, for each table 5 files will be generated total 25 files generate for 5 tables and these files will copy from publisher database to snapshot folder.

4. Once snapshot generated automatically locks will be removed on objects and breaks connection between publisher and distributor.

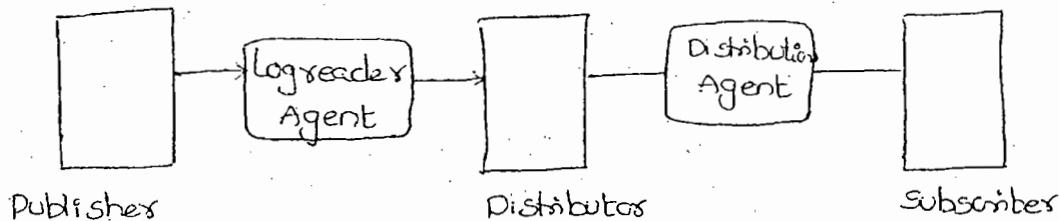
5. Distribution agent works from here. It goes to snapshot folder identifies tables and creates same files in subscribers. Once files copy done to subscribers the relation between distributor and subscriber also breaks.

When we go for snapshot

- (a) When source database has constant data
- (b) When huge latency is accepted.
- (c) When we do not want changes in subscriber regularly.

Transactional Replication

Transactional replication is one way replication. Initial synchronization will be done using snapshot replication. Snapshot generation process is same in all types of replication. on going changes should be tracked and copy those changes to subscriber continuously using Log reader agent.



In transaction replication changes will be tracked through log files. While copying modifications locks will be applied once schema is copied locks will be released, Log reader will read on going changes by scanning log file of the database. From here distribution agent will copy data to subscriber. This we call as "concurrent snapshot".

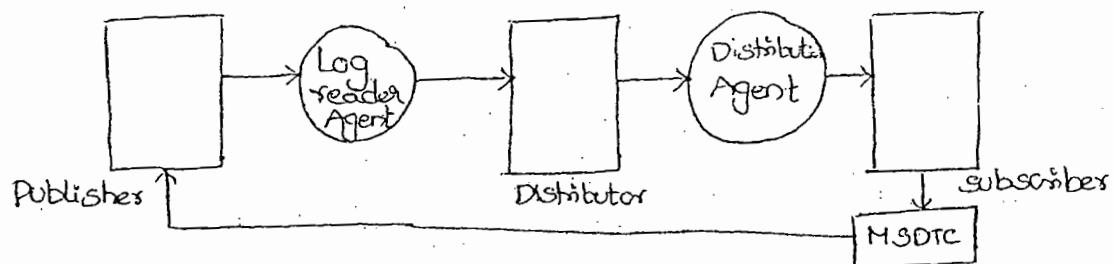
Transactional with updateable subscriptions

Updateable subscriptions is Two way replication. Any changes on subscriber will be moved to publisher. There are 2 types of updateable subscriptions.

(a) Immediate updating

(b) queued updating.

Immediate updating uses MSDTC (Microsoft Distributed Transaction Coordinator) to apply a two phase commit transaction. The triggers on subscriber use MSQC to call stored procedures on publisher to update the data.



Queued updating

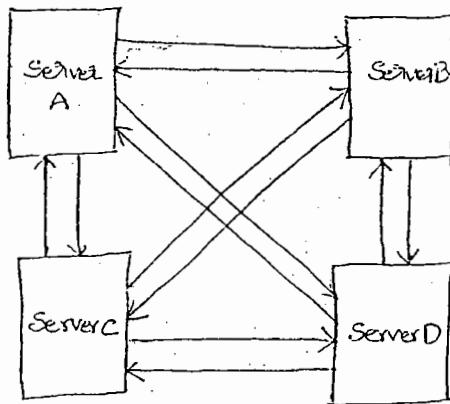
Queued updating copies changes to queue Msreplication_queue. There is another agent queue reader agent which reads transactions from Msreplication_queue and moves to Publisher later time.

When transaction commit on both servers then it treat as committed transaction.

If it commit only one server, it treat as failed transaction, MSDTC will rollback these transactions.

Peer to peer replication

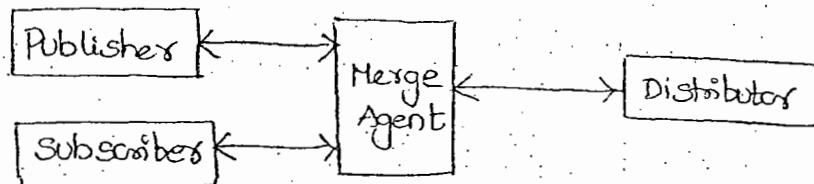
Peer to peer introduced in sqlserver 2005 and enhanced in sqlserver 2008. Each server will act as a publisher and a subscriber. Data will move to all the servers.



We have conflicts when one server fails in this replication. In sqlserver 2008 we can add or remove servers without interrupting the other servers. Conflicts handling introduced in sqlserver 2008.

Merge replication

Merge replication allows publishers and subscribers to make changes to data independently and then merge the results. We have conflicts with merge replication. It uses Merge Agent.



Configuring Replication

In any type of replication we need to configure distributor first then publications and subscribe those publications.

Pre-requisites in replication:

Tables participate in replication has primary key then only

We can configure Transaction replication using them.

Under Server Management studio of Distributor

1. Right click on Replication → Select Configure Distribution
2. Provide server which used as Distributor. Click Next
3. Snapshot folder screen appears, provide path
4. Distribution Database will ask details, provide
 - (i) Name of the distribution database
 - (ii) Folder for distribution data file and log file
5. Publishers screen appears, here we can see and Set the list of servers that can use this distribution database.
6. Wizard Action screen provide option to configure distribution distribution or Create script.
Select both options and click Finish
7. Wizard Actions completed successfully. If any errors reported investigate and resolve them.

After successfully configured distribution.

Open system databases → you able to see new distribution database and new jobs created for replication.

Step 2 : Configuring publisher

1. Connect to publisher server , Expand Replication
Right click Local publications → select New publication
2. on publication database screen → select database for publication
3. on publication type screen → select type of replication (Transactional publication) click next.
4. on Articles → select the tables you want to replicate
5. on Filter Table Wizard will give option to filter out rows
6. Snapshot Agent screen , When snapshot should be scheduled to run
 - a. Create snapshot immediately (Manual)
 - b. Schedule snapshot agent to run (Automation)
7. Provide snapshot agent service account to connect to publisher
8. Provide publication name and click Finish.

E Configure subscription

1. Connect to subscription server → Right click Local Subscriptions and select New subscriptions.
2. Browse publisher Name → select push (or) pull subscription

Push → Distributor pushes data to subscriber

Pull → subscriber pulls data whenever required to subscriber.

3. Subscribers screen → set subscriber properties

(We can add additional subscribers from ADD if we want)

4. Distribution Agent security screen → provide security setting for connecting

5. Specify synchronization schedule → select either Run continuously (or) Run on demand only.

6. on Initialize Subscriptions → select initialization immediately (or) After some time.

7. Click Finish

Replication Monitoring

Select Replication and Right click on it in ssms

→ Select option Replication Monitor , It launches replication monitor on the Server.

We can get details of Latency between publishers and subscribers, data flow between Publishers to distributor and distributor to subscriber will display.

CLUSTERING

clustering is a windows level high availability option to increase the availability of physical server hardware operating system and sqlserver instances. If any of these 3 fails on one server, the other server in cluster automatically takes tasks of failed sqlserver instance.

A windows failover cluster can support upto 16 nodes in sqlserver 2012. clusters reduce downtime to minimum. Most clustering deployment is between 2 servers only. The clustered servers called as Nodes.

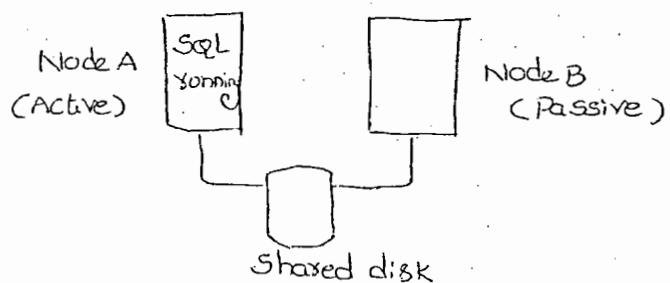
How Clustering Works

A single sqlserver 2012 instance can run on only a single at a time. In 2-node clustering, the server on which sql services running called as Active node. Other server on which services not running called as Passive node. If active node fails passive node becomes active and continues user requests of the server. Active node access the database on shared disk and sends heart beat signals over private network and passive node receives them. If active node stops passive node notices that heart beat stopped. Passive node assumes that active node has failed (129)

and initiates a failover. Passive node now takes over the control of shared disk.

Sqlserver cluster assigns own virtual name and virtual IP address. These details shared by both nodes. Users connects to cluster using this virtual name. If a failover occurs, the cluster will still retain same virtual name and IP address although a new node will be responding to user requests.

SQL Server clusters



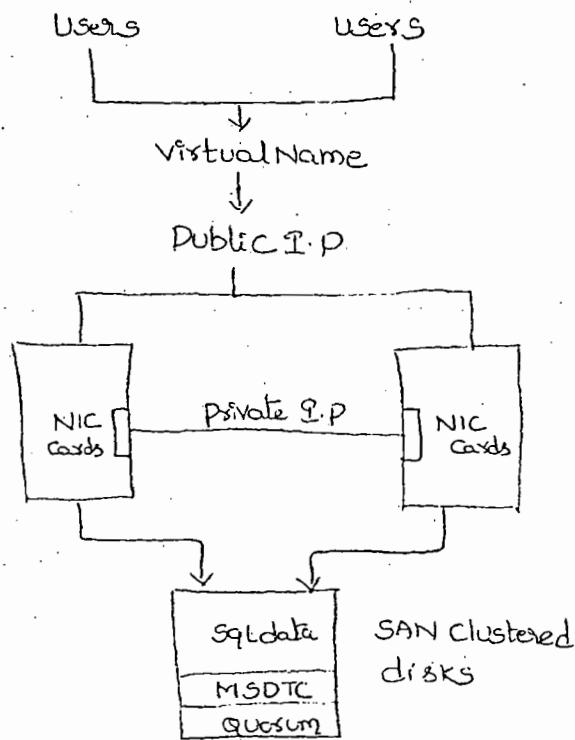
Advantages

1. Cluster reduces downtime in upgrading hardware (or) install a new service pack on sqlserver.
2. Clustering provides high availability to physical server hardware, operating system and sqlserver.

Disadvantages

1. Clustering requires expensive hardware and software.
2. Geographically dispersed clusters requires SAN hardware and software for data transfer.

Architecture of clustering



Virtual name is shared by both servers, Active node connects to storage disk and access SQL databases, MSDTC and Quorum. Transactions goes to whichever server is Active, when heart beat signal between NIC fails automatic failover happens using internal heartbeat mechanism.

Types of clustering

1. Active - Active Cluster
2. Active - Passive cluster.

Active/Active cluster is to make better use of available hardware. Instead of keeping passive node idle, there

are two instances, one on each physical node of cluster.

To save hardware costs we can use active-active clusters.

When a fail occurs 2 instances will run on single server

Each instance needs its own logical shared disks. The disks of 2 sqlserver instances will be different, disks cannot be shared among sqlserver instances.

When 2 instances run on single server, the performance may effect. We need to configure both servers with high configurations to meet Active-active cluster requirements.

Active -passive cluster.

The node sqlserver instance running is called as Active node and other server sqlserver not running referred as Passive node, it is in a state of readiness.

If failover occurs Passive node automatically becomes active and starts serving user requests, formerly active node becomes passive node.

Here both nodes connect to external storage disk, only active nodes access the sql databases HSDTC and quorum.

Components required to configure clustering.

1. virtual servers
2. Nodes
3. shared disk
4. quorum
5. public and private Network
6. MSDTC

Virtual Name : Both of the servers is using virtual name of cluster. User connects to SQL cluster using this virtual name and IP address. Primary node will respond to user requests. If a failover occurs to secondary node still cluster will retain same virtual name and IP address. Users may not notice that failover happens inside the cluster.

Nodes

Nodes referred as physical servers in clustering. These nodes have same access to shared storage. Each node talk to one another via a network. If one node does not communicate to other node in cluster, the secondary node will take ownership of user transactions.

Automatic failover : one server stops, other node becomes owner of cluster.

Manual failover : DBA manually failover for maintenance

Shared Disk

Clustered instances store data on shared disk. Both nodes physically connect to disk but only active node can access the instance databases. This shared disk have at least 2 logical partitions. One partition is used for storing the clustered instances SQL Server databases and other is used for quorum.

Quorum

Quorum is a log file similar to log file, records any change made on active node, when passive node becomes active it reads quorum log file and applies changes on newly active node. This file quorum.log resides in the quorum disk. Quorum is main interpreter between nodes. It stores latest cluster configuration and helps other nodes to take ownership when one node fails.

Voting Mechanism

Quorum is designed to handle split brain scenario when nodes unable to communicate each other, each node assumes and bring resources online. When same SQL resources brought online on more nodes at same time lead to data corruption, this scenario is called split brain.

Example

We have 4 nodes, 1 instance of SQL running on each node. Node 1 & 2 lost communication with Node 3 & 4. In this scenario Node 1 & 2 try to bring SQL instance resource owned by Node 3 & 4 in the same way Node 3 & 4 will try to bring SQL resource owned by Node 1 & 2 which will lead to disk corruption and other issues.

Quorum is designed to handle this scenarios

The cluster will force the cluster service to stop in one of the nodes to ensure only one owner for a particular SQL Server resource group

Voting Process in Quorum

Quorum is based on voting algorithm. Where more than half (above 50%) must be online to communicate each other. Cluster knows how many nodes available. Each node will have vote, if number of votes drop below majority cluster service will stop.

Cluster requires more than half of total votes to achieve quorum. This is to avoid tie between nodes. In 3 node cluster 2 voters must be online to have quorum. To achieve majority votes we have 4 quorum models (135)

1. Node Majority
2. Node/Disk Majority
3. Fileshare
4. No Majority

Public and private Network

Each node of a cluster must have at least 2 network cards. one network card will be connected to public network and other to a private network.

Public network is the network where users are connected and communicate to clustered SQL Server instance.

Private network is used for communications between the nodes of cluster. It is used mainly for what is called the heartbeat signal. 2 forms of communications are executed

Looks Alive : verifies whether SQL Server service runs on the online every 5 seconds by default.

ISAlive : verifies that SQL Server accepts connections by executing SP-SERVER-DIAGNOSTICS command

MSDTC

Microsoft distributed transaction coordinator, MSDTC is used for distributed transactions between clustered nodes and other remote servers. If transaction in a query have distributed transactions, we need MSDTC as cluster resource.

Clustering Administrative Tasks

Failover cluster Manager

Failover cluster Manager

SQL 2012 cluster (virtual Name)

Services and applications

Sql Server

MSDTC

Nodes (Physical servers)

Sql 2012 Clu 1

Sql 2012 Clu 2

Storage

Networks (Public & private NW)

Cluster Network 1, Cluster Network 2

1. Moving services from one node to second node

Right click on services under services and applications

→ Select → Move to this (Node 2 or Node 1 select)

2. Stop cluster service

Right click on the Node → Go to More actions

→ Select stop cluster

(We will have start cluster option enabled once we stop cluster.)

3. Take service offline / online

Right click on sql server under services and applications

Select → Take this service (or) application offline

it moves services to other Node

4. Add storage

Right click on sql server → Select Add storage

We can add new drives to cluster storage

We can move resources in Add a resource

5 Remove a Node from cluster

Select a Node under Nodes section → Right click

on it → Go to More options → Select Evict

option it will remove Node from clustering

6. Shutdown / destroy cluster

When clustering no more required on this server

We can use this option.

Select virtual server cluster name → go to more actions → select shutdown cluster (or) destroy cluster

Configuring Windows Clustering

1. On Window server → Go start → click Administrative Tools
→ click Server Manager → under features
 Select Failover clustering option
 and click Install

Complete the installation process on 2 Nodes

2. Open Failover clustering on start → Administrative Tools
→ Failover Cluster Manager
Validate the Nodes and shared disk → Browse and
select servers in cluster → Run tests
3. Create cluster → Add names of servers that we want
to have in cluster by clicking browse →
 Type the Name for cluster Name (Virtual Name)
 and IP address

4. Shared disks → Add drives to cluster from window
5. on confirmation windows will display cluster summary.
(i) cluster name and IP address
(ii) Nodes details
6. Click Next it will start creating cluster, once
it successful completes cluster installation displays
Summary
7. Once fail cluster manager we can see Nodes & storage.

Configuring sql clustering 2012

1. Click on New SQL Server fail over cluster installation in SQL Server Installation screen
2. Set up support files installs and gets License key from Software
3. In Instance configuration window provide SQL Server Cluster Name (virtual Name)
This name will be available for connecting to sql server
4. Under Cluster resource group → provide details what are components need to be clustered.
5. In Cluster disk selection → select available disk groups for new SQL 2012 cluster instance
6. In the Cluster Network Configuration provide IP address of cluster for public network and Private Network
7. Provide cluster services account details in cluster security policy.
8. On the complete screen check for successful installation
① Your SQL Server 2012 failover cluster installation is Complete.

ALWAYSON

AlwaysOn is new high availability - disaster recovery feature in SQL Server 2012. Initially it named as HADRON.

HADRON - High availability Disaster Recovery Always On
AlwaysOn uses clustering technology for failovers and also Mirroring in keeping multiple mirrored copies of the database.

In Always on, we can failover multiple databases at the same time using Availability Groups.

→ AlwaysOn provides ability to have multiple replicas which is not available in Db Mirroring. We can have one Primary and upto 4 secondary replicas with alwayson.

The secondary replicas configured for read-access
We can use for reporting purpose, backups etc.,

→ Always also have 2 Availability modes

i) Synchronous

ii) Asynchronous

Synchronous provides zero data loss

Asynchronous provides better performance, chance of data loss.

For each replica we can configure separate mode

Components of AlwaysOn

1. Availability Group
2. Availability replica
3. Primary replica
4. Secondary replica

Availability group is a container used to group databases. AlwaysOn Availability groups contains multiple databases all of which can be automatically failed over as single unit.

Availability replica, is an server that contains the Availability group which can use for failover.

Primary replica is a server the holds read-write copy of the databases contained within Availability group.

Secondary replica is secondary server the holds copy of Primary replica databases, only read option is available here in an Availability groups.

Always on SQL Installation

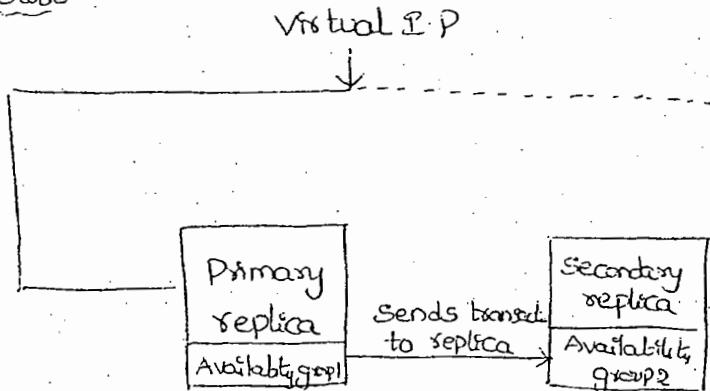
AlwaysOn does not require cluster installation. We can go through normal SQL server installation. Here entire SQL instance will be not be clustered only availability

which participated in AlwaysOn will be clustered. We need to maintain same drive letters and paths for databases on replicas.

After installing sql server Go to Configuration Manager → MssqlServer → Right click and go to Sqlserver properties → click AlwaysOn High availability. We will get Windows failover cluster Name as part of Windows Server failover cluster installation. Under that → enable AlwaysOn Availability Groups.

1. Once we enabled AlwaysOn Availability Groups we can create Availability groups that contains all databases which we want to failover.
2. Go to SSMS → Right click on Availability folder → It opens wizard → Select the databases which you want to keep in AlwaysOn in Primary replica as well as replicas that will host on Secondary replica here.
3. The wizard automatically creates end points to synchronize data, and takes backup of primary replicas and restore them on secondary replicas.
4. If we don't want we can do Backup and restore manually.

Architecture



Always on availability group is superior to Sql server clustering

- Configuration, deployment, Installation is relatively simple Comparing to normal clustering
- All replicas (Nodes) will have a copy of databases there is no shared storage hence we can avoid single point of failure
- We have readable secondary replica hence we can distribute read-only load to secondary node and Read-write load to primary hence better utilization of both servers. Whereas we have to keep in passive state in clustering. There is no concept of readable databases as we keep everything in shared disk.

AlwaysOn Superior to Database Mirroring.

- We have upto 4 mirrored instances replicas here.
- We can use combination of synchronous commit mode for some databases and Asynchronous commit mode for other databases at the same time; not possible in Mirroring

→ Synchronous mode for high availability (Automatic failover).

C and Asynchronous mode for disaster recovery purpose.

→ We can combine 2 or more databases and failover them as a unit, we can perform this db mirroring.

→ Databases are always in recovery mode in Mirroring here secondary replica databases are readable, hence we can take load from primary replica.

→ Backup operations can be performed on secondary replica databases.

How Always On Works

Always on featured based on windows cluster services only. Availability groups are a grouping databases with a virtual name and IP address that act as a single unit for users to access. If a server fails within an availability group the entire group is failover to the secondary replica of secondary server.

To create databases we need sql server to be installed on a server under windows clustered. SQL Server clustering does not need in Always on.

Primary replica sends Transaction log records of each primary database to every secondary database. The (145) secondary replica applies changes on secondary database.

Data synchronization occurs between primary and secondary databases. Windows failover clusters monitors this resource group to evaluate health of primary replica.

The quorum for AlwaysOn Availability group is resides on all nodes of cluster which helps in failover. There is no witness role in AlwaysOn Availability groups.

Types of failovers are similar to mirroring.

Synchronous commits supports planned and manual failover. Automatic process and no dataloss.

Asynchronous commits supports only forced failover. chance of dataloss.

Advantages

1. We can maintain multiple secondary databases.
2. Secondary database used for read only access, we can use for reporting purpose and backup's.
3. Protects data against disk failure.
4. Automatic failover and group of databases failover as single unit.

Disadvantages

1. Requires more disk space to store secondary replica data.
2. Requires special setup configuration.
3. Expensive to implement.

PERFORMANCE TUNING

Performance Tuning Process consists of identifying Performance bottlenecks, prioritizing the issues, troubleshooting their causes, apply different resolutions and Quantifying Performance improvements.

Issues

1. Query Execution taking Long time
2. Performance of the site is slow.

Performance Tuning involves following check list.

1. Minimizing blocking and deadlocks.
2. Analyzing Resource consumption on server level.
3. Data design analysis
4. Query design analysis
5. Execution Plan analysis
6. Indexing maintenance and Fragmentation analysis.
7. Using Database Engine Tuning Advisor
8. Isolation Levels of a query.

1. Locks, Blocks and Dead Locks

For most of the performance issues the source is locks.

Definition :- Locking is a mechanism which provides consistency as well as concurrency to data and database objects. Lock makes other process to wait until first process complete.

Consistency → Accurate results

Concurrency → Simultaneously accessing database objects for multiple users.

Developers intentionally keeps locks on database objects to avoid inconsistent results.

Ex :- on a Table one user updating data, other user reading the updating data, he gets inconsistent results while reading data which is updating.

To avoid this we are serializing the read and write operations. Read and write will perform one after other until write operation completes internally applies lock on objects, this lock keeps other user to wait until first process completes.

Concurrency means read and read can perform same time. Select operation allows multiple users

Types of Locks

1. Shared Lock denoted by letter S
2. Exclusive Lock
3. Update Lock

Shared Lock

Shared lock will be acquired with a select command. When a select command runs on a table, it applies shared lock means takes ownership on table. Shared lock allowing sharing access to other select operation. If other user run update it will not allow until shared lock is released. We achieved consistency as well as concurrency. Multiple users can access at a time. Duration of shared lock is till the completion of select operation.

Exclusive Lock

Whenever any updation, insert (or) deletion performs on a table, Sql Server is going to apply exclusive lock. Once it got exclusive ownership on object, it will not allow any operation either read (or) write.

Ex :- In logshipping; when we keep database in stand by mode we are selecting "Disconnect user while starting restoring". here restoration applies exclusive lock. It supports only one process at a time.

Consistency is very high, concurrency is less.

Duration of lock is till entire transaction completes.

Update Lock (U)

Update lock is combination of shared lock and exclusive lock.

Ex :- updating any record performs 2 operations. first it will searches the record and then it updates data.

If update lock applies on a object, till select operation performs it applies shared lock on transaction, once select completes, when actual updation happens it applies exclusive lock means converts shared lock to exclusive lock on object. Concurrency is more here.

Intent lock

This is different from other 3 locks. It is acting just like a table which stores information of locks on transactions. Sql server came know that a lock is applied on table using intent lock.

We can see locks by using

- (a) SP-Lock, which is old mechanism
- (b) From 2005 onwards, we have latest mechanism
Select * from sys.dm_tran_locks

LOCK Escalation

Converting the locks held on Low Level to High level is a process of Lock Escalation. We can start applying locks on rowlevel, pagelevel, tablelevel and finally databaselevel.

Ex:- on a page several records store inside page while updating table, if i keep a lock on single record it's called Row Level Lock. When ever there more row locks applied sql server internally converts all row locks into a single page level, this process is called Lock Escalation. Every lock consumes 5kb of memory.

SQL Server automatically triggers the lock escalation in 2 situations

1. When ever SQL Server running out of RAM, the minimum level locks automatically converts to next level by releasing all low level locks it applies one next level lock
2. When ever it reaches threshold of 4000 individual locks on low level, sqlserver triggers next level lock.

Isolation Levels

Isolation level is a property of transaction that can set the locking behaviour.

Locks limited to statements, For applying lock on entire transaction we are using property of ACID called Isolation.

How to set Isolation, After writing transactions we can write on top of transaction like

```
{  
    Set Transaction - Isolation Level 'Isolation Level name'  
}
```

Types of Isolation levels

Till SQL Server 2000

1. Read Committed
2. Read Uncommitted
3. Repeatable Read
4. Serializable

From SQL Server 2005 onwards we have 2 more

5. Snapshot Isolation Level
6. Read Committed Snapshot

Why we need these Isolation Levels. This is to eliminate inconsistency. There are 4 chances of inconsistencies

1. Dirty reads
2. Nonrepeatable reads
3. Lost updates
4. Phantom reads

Dirty read, if there is no lock on object each and every read output is changing and gives different results.

Nonrepeatable reads, same as dirty reads output gives different results means output is not repeatable.

Lost updates, when 2 updates are performing on same time on same objects, finally updates got finished and see results with second update. first update is completely missing, here one update is lost.

Phantom reads, updating one record, there may be insert and deletes on other records here data may be consistent.

We need to eliminate these inconsistencies using Isolations.

Read Committed

Reading the fully committed data, here shared lock on the transactions will apply. Modifications will not allow.

We can eliminate dirty reads and Nonrepeatable reads.

Read Uncommitted

No lock applies in this Isolation. It allows all 4 inconsistencies.

Repetable read

Keeps updatable lock on transactions, initially it keeps shared lock and converts to Exclusive lock while updating. We can eliminate dirty reads and non repeatable reads here but a chance of Phantom reads.

Serializable

It keeps exclusive lock, not allows any other process. Eliminates all 4 inconsistencies but users have to wait more time for lock to be released. Concurrency is low.

Snapshot Isolation

Before applying exclusive lock, it copies table to Tempdb, if any user access data he can get results from tempdb. Once exclusive lock completes modifications it breaks tempdb and provides committed data results. We get both consistency and concurrency.

Read Committed Snapshot Isolation

BLOCKING

When one process is forced to wait because of other process is called blocking. First process is blocker and second process is blocked process.

Ex:- on a Table, one transaction is running using exclusive lock, until this complete second process has to wait. Transaction one is blocking transaction two. This long time waiting to release lock is called blocking.

Reasons for blocking

1. Lock compatibility.
2. Shortage of resources.

Lock compatibility, Shared lock is compatible with other shared lock, whereas exclusive lock not compatible with any other locks because of this other process are forced to wait, mainly blocking happen because of this lock compatibility.

Shortage of resources, one transaction consumed all CPU resources and memory resources on a server, second transaction is waiting for other resources to release resulting blocking.

When ever system is slow, queries taking more time usually users complain then dba's come to know blocking. We configure blocking alert when it reaches threshold.

Identifying blocking

1. Select * from sysprocess where blocked > 0

There is a column named 'blocked' in sysprocess table.

It identifies blocking is there or not. If blocking is there it will give information about process involved.

2. SP-WHO2, it shows all records which is currently running.

3. Select * from sys.dm-execute-requests

This DMV gives indepth details, it gives internal metadata including resource consumption, memory consumption etc.

4. Select * from sys.dm-os-wait stats

Provides statistical information regarding why waiting happens.

5. Select * from sys.dm-os-waiting tasks

What tasks responsible for blocking we gets by this DMV.

We can use any of the above 5 process to identify the blocking. We can obtain blocking information from Activity monitor also. It gives blocked spids, head of blocking chain, how long process has been running.

Steps for Troubleshooting blocking

1. Identify are there any blockings on server.
2. Identify what are the processes involved in blocking.
3. Identify root cause, need to check for Head blocker.

Ex :- i) SPId 52 is blocked by 56 then 56 is head blocker

(ii) SPId 52 blocked by 56, 54 blocked by 56, 56 is blocked by 58 here 58 is main/head blocker

4. Find out what exactly blocked and blocker process is doing by using DBCC input buffer (spid)

Ex:- DBCC input buffer (60)

DBCC input buffer (80)

Here 80 is blocked by 60. If 60 is select operation we can kill directly as there will not any rollback. If it is a insert or update or a stored procedure, gather all the information and contact Application team.

5. Need to identify user, login of the process, from which server this process is running.

Select * from sys.dm_exec_sessions where session_id=80

It gives detail information of login, host user.

6. Provide all these details to Application team, once we get approval we can kill any process using kill(spid)

Dead Locks

Dead lock is a special blocking scenario in which two processes get blocked by each other, here the 2 processes mutually blocked and not able to continue further.

Scenario 1 :

Two processes has a shared lock on a resource, each processes attempts to promote lock from shared to exclusive, however neither can proceed further until the other releases its shared lock this leads to deadlock.

Scenario 2 :

Two transactions are running T_1 consumes total free memory, because of poor designing of query T_1 went to sleep state and never comes to end. T_2 is waiting for that memory to release. Leads to deadlock.

Handling Deadlock

Sqlserver regularly checks for the presence of dead locks, inside sqlserver there is a process called LOCK MONITOR, it scans each and every lock every 5 or 10 seconds. When ever lock monitor suspects sign of deadlock, it minimizes the lock interval.

to very minimum, even to milli seconds. and kills one of the process based on dead lock priority and faster roll back transaction will be killed by sqlserver. and it becomes dead lock victim.

Choosing Dead Lock victim

There are 2 properties to kill transaction in sqlserver

1. Dead lock priority
2. Faster roll back.

Dead lock priority, Developers set priority while designing transaction, either High, Medium or Low. In case of dead lock situation sqlserver gathers complete information about 2 transactions. based on priority it kills one transaction which is low priority.

```
SET DEADLOCK_PRIORITY LOW;
```

High priority transaction never becomes dead lock victim in case of both have same priority then it moves to other property in choosing victim.

Faster roll back, If T1 takes 30 minutes to roll back and T2 takes 3 minutes, then sqlserver kills T2. (159)

If both have same time in roll back then sqlserver randomly chooses one transaction and sends notification to user as "Your Transaction rolled back by Dead lock, re-run"

Prevent dead lock in future

The Role of DBA is to prevent deadlocks in future by gathering deadlock information. We make configuration in sqlserver to log deadlock errors in sqlserver logs. We need to enable trace flags to get deadlock info.

How to enable Trace flags

There are 2 ways ① DBCC Traceon ② startup parameters

DBCC TRACE ON (1204)

Trace flag 1204 captures deadlock information from lock monitor in Text format.

1222 captures in XML format.

3605 Whatever lock monitor captures it writes that information into Sql server logs.

From Configuration Manager → select services
Right click on it and go to Advanced → under
Startup Parameters add trace flags as -T1204,
-T1222, -T3605.

2. Resource Consumption

Performance problems in sql server can be caused by more resource consumption on server level.

We have 3 types of resource consumptions

1. High CPU utilization
2. Memory
3. Physical Disk IO

CPU Utilization

Whenever we have high CPU utilization on sever, we have various tools to monitor them

- (a) Task Manager.
- (b) Activity Monitor.
- (c) Performance monitor (Perf Mon).
- (d) SQL Profiler.
- (e) Dynamic Management Views (DMVs)

Task Manager

Whenever CPU usage is over 80%, the first tool we need to look is Windows Task manager.

Go to Task Manager → click on process tab.

Verify the CPU usage is due to sqlserver (sqlservr.exe) activity or it is caused by other process.

Activity Monitor

From sqlserver 2008 Activity monitor is introduced to monitoring sqlserver.

To Launch Activity monitor, right click on sql server Select Activity monitor. once it launches we will see ~~4 graphs~~ on screen.

1. Processor Time Waiting Tasks Database I/O Batch requests

under these we have 4 categories.

Processes

Resource Waits

Datafile I/O

Recent Expensive queries.

The graphs on the top provide brief history about the Performance of server. While the below 4 categories provides more information about what consumes most CPU resources on the server.

Performance Monitor

This windows monitoring tool is capable of measuring the cpu usage by sql server.

From : Control panel → system and security → Administrative Tools

● SQL Compilations, for any query sql server generates a execution plan in procedure cache. compilations utilize more amount of resources . It should be less than 10% of batch requests/sec.

SQL recompilations should be less than 10% of number of SQL compilations/sec.

Using SQL Profiler

Profiler is a tool that captures all server data, captures blocking and deadlocks ,

From SSMS → Go to Tools → Select SQL Profiler and → Connect.

Events mostly used for CPU usage monitor is RPC completed , whenever any request comes as stored procedures we need to capture details about the stored procedure execution after it completes.

SP Completed , capturing manual execution of stored procs

SP recompile for excessive recompilations details

SP: Statement completed ; Whenever particular statement inside a stored procedure is completed we will capture that statement.

The primary Performance counters for monitoring CPU are

i. Processor Time

Processor queue length

ii. Privilege Time

These 3 counters are sufficient to monitor overall CPU

usage. However we have SQL Server counters that monitors such as compilation and recompilation which can consume lot of CPU.

Batch Requests/sec

SQL Compilations/sec

SQL Recompilations/sec

Processor time identifies how much time CPU is busy in executing SQL operations. It has to be less than 80%.

Processor queue length identifies number of threads waiting on processor resources for processing. It has to be < 2 per processor.

Privilege time identifies system process on server. It has to be less than 30%.

Batch requests/sec, How many transactions SQL Server currently executing per sec. It has to execute 1000 batches/sec

DMV's

To identify high CPU consuming we use DMV

1. Select * from sys.dm_exec_requests order by CPU_time

Select * from sys.dm_exec_requests

Shows each executing request inside the SQL Server

We need to verify CPU-time column and identify which process consumes high CPU.

2. sys.dm_exec_sessions

Verify CPU-time column and check which process consumes CPU.

3. sys.dm_exec_query_stats

Inefficient query plans can increase CPU consumption

Verify them using this DMV.

4. sys.dm_exec_query_optimizer_info

We get details of how much time SQL Server spent on optimization. In this check elapsed time and final cost columns.

Monitoring Memory Issues

When SQL Server does not have enough memory other resources such as CPU and disk IO are often impacted.

SQL Server uses memory to process requests. Whenever request process it consumes memory and once request it releases the space for other process.

How to identify insufficient memory

There are 2 types of Memory 1. Physical Memory and 2. Virtual Memory (VAS). Physical memory is amount of RAM present physically on system for all operations. Virtual memory, space required for processing internal operations on OS (Kernel Level).

Memory Pressure :- When there is insufficient memory we are calling as Memory pressure.

There are 2 types of Memory pressures.

1. External Memory Pressure

2. Internal Memory Pressure.

External Memory Pressure, whenever operating system having shortage of memory we called as External.

Internal Memory Pressure, whatever we have given to SQL Server is not sufficient we call as Internal.

When ever system identifies external pressure , it trying to trim the currently running processes which can results overall slowdown of the system . When sql server detects this condition sqlserver starts cleaning internal caches to release some space in Buffer pool using LRU (Last Recent Used) Algorithm.

Internal pressure , When ever sql server identifies high memory consumption it going to perform redistribution of memory between internal components Buffer cache, Plan cache , Logs, Latches and locks .

Situations of Internal Memory Pressure

1. Heavy work load on SQL server .
2. Due to External pressure , sql server releases its own memory which impacts internal sql memory .
3. When ever some one change memory settings like Max server memory and min server memory it leads to sql server internal memory pressure .

When ever system is running on virtual memory and that is low , we have to configure page files for space and "Mem to Leave" means leave sufficient memory to operation system for its operations .

In case if you identify Memory pressure

1. Go to Task Manager → Right click and select Performance Tab to see what exactly is consuming high memory. What is the physical memory usage and what is the Kernel (virtual) memory usage. If it is Non-SQL operations need to check with that team.
2. We need to perform proper analysis whether allocated RAM is sufficient or not. We have to request for additional RAM.
3. If it is SQL Server operations check memory setting whether they configured correctly or not. Need to verify indexes design, Batch requests coming to SQL Server, identify problematic queries which consumes SQL memory.

Tools to identify for Memory issues

When ever we have time out errors due to shortage of memory, we have to analyze the root cause for permanent solutions.

- (a) Dynamic views (DMV's)
- (b) Performance monitor

Using Performance monitor counters

~~These are 2 types of counters objects~~

1. Memory , memory given to operating system
2. SQL Buffer, Memory given to SQL server.

Memory : Available bytes , How much memory still available in operating system . It should be always high , Less than 20 MB is insufficient RAM on system.

Memory : Pages/sec

How many Pages written to disk and how many Pages are getting from.. No of Paging when os RAM is Less, allowed is Less than 50 per sec.

Object for sql server is SQL Server Buffer Manager

Page Life Expectancy (PLE)

Once any Page copied to Buffer , it has to stay atleast 5 minutes, if it is Less we can say there is issue in buffer.

Buffer Cache hit ratio

How many Pages read from Buffer and how many read from disk , The ratio between them Less than 90% indicates Problem with buffers.

Total Server Memory and Target Server Memory

(69)

Memory allocated to sql server is total server Memory , How much sql is looking called Target . Total has to be > Target

DMVs for memory related issues

1. sys.dm_os_sys_memory
2. sys.dm_os_process_memory
3. sys.dm_os_memory_brokers
4. sys.dm_exec_requests
5. sys.dm_exec_sessions

OS-SYS-Memory DMV is capable of displaying, how much physical memory is consumed by sql server we can identify by columns.

Physical-memory-in-use, how much memory physically allocated will be displayed.

sys.dm_os_Process_Memory

In the output we get details like Physical-memory-in-use, Large-Page-allocations, Memory-utilization-Percentage

sys.dm_os_memory_brokers

IO Issues

SQL Server performance depends on disk IO subsystem.

We need to identify whether disk is over utilized or not.

We have performance counter to identify IO issues.

Avg. Disk queue length.

The queue length represents average number of read and write requests that are queued on physical disk.

If queue length value exceeds 2 might have IO issues.

% Disk time

How much the disk is busy in performing SQL operations.

Oversusing disk may lead to disk failure. The usage always less than 50%. If more than 50% try to split usage by adding more disks.

Avg. Disk reads/sec

While doing read operations it has to complete in 8 milli sec. If it taking more time we have to raise request to storage team.

IO issues identifying using DMVs

1. Select * from sys.dm_io_pending_io_requests

Provides details of each pending IO request.

IO-Pending column gives pending details.

2. sys.dm_io_virtual_file_stats

This DMVs provides details about read and write operations

num-of writes, num-of reads, size on disk

3. Resource Monitor

Go to Task Manager → click Resource Monitor

Select Disk tab

Resource Monitor

Disk

Process with Disk Activity

Disk Activity

Storage

under Disk Activity we get details about current processes using disk. The Response Time (ms) is useful to see how long it is taking to service that I/O request.

SQL PROFILER

To find out Long running queries using SQL Profiler.

From SSMS → Tools → SQL Server Profiler

General	
Trace Name	SQL SP Test
Trace Provider Name	SQL2014-PC\MSSQL2012
Trace Provider type	Denali version 11.0.1400
Use template	standard default ▾
<input type="checkbox"/> save to file	
Set Maximum file size (MB)	5
<input type="checkbox"/> enable file rollover	
<input type="checkbox"/> save to table	
<input type="checkbox"/> enable trace stop time	Run

TraceName → Trace records all data inside sql server
We need to provide name.

Trace Provider Name → Server name on which we are
running profiler, it displays sql name and its version.

Use template → By default we keep as standard
First time it captures the data of all the events selected.
We can use this events as template for next time run.

Save to file or Save to table → we can store the
trace in the form of file or Table.

Set Max file size (MB) → By default it will be 5 MB
 Whenever it reaches file size of 5 MB, Profile trace
 will starts a new file

Enable trace stop time → Automatically we can stop
 the trace on mentioned time using this option

Events selection	Application Name	LoginName	cpuseconds	clientprocess
Sessions	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Stored Procedures				
<input checked="" type="checkbox"/> RPC completed	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
TSQL				
<input checked="" type="checkbox"/> SQL Batch completed	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/> SQL Batch starting	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
LOCKS				
<input checked="" type="checkbox"/> Lock Escalation	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/> Dead lock graph	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/> Lock Time out	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Performance				
<input type="checkbox"/> Degree of Parallelism		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/> Show plan ALL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> Show plan XML				
<input type="checkbox"/> Show all events		<input type="checkbox"/>		
COLUMN FILTERS	ORGANIZE COLUMNS			RUN

Events selection

We are specifying to capture particular transactions based on conditions we need to restrict the data capturing.

Columns → what data we need to capture exactly.

Mostly used events

1. RPC completed

Whenever any request comes from front end it will connect to database will come as stored procedures. once stored procedure execution will be remotely we call this as Remote Procedure call.

2. SP completed

Executing stored procedures manually will be captured in backend.

3. SP recompile

What query is going for excessive recompilations that details of stored procedures will capture here.

4. sp: statement completed.

Whenever particular statement inside a stored procedure we will capture that statement details.

5. sql batch completed.

Whenever one batch is executed we need to capture who run it, what is login, cpu we will capture here.

6. SQL : statement recompile

Recompilations of statements data need to capture here.

7. Lock escalation.

Whenever any lock applied on objects it captures data.

8. Dead lock graph

Captures all data sequentially in graphical info in dead lock chain.

9. Lock time out

When query got timed out we are asking to capture data.

From event selection we can check what is application
the query is running , what is CPU consumption
under which spid it is running , Duration , Hostname

We will capture all these details for further
investigating of long running queries.

Column filters

We have to give on which database we need to trace

on Numerical columns Equal to Not equal to

Greater than Less than

on Text columns Like Not like

We need to organize columns up and down here.

click RUN .

INDEXES

Index is a pointer which points existing stored data.

Indexes play very important role in increasing response time of query and it increases Performance.

Types of Indexes

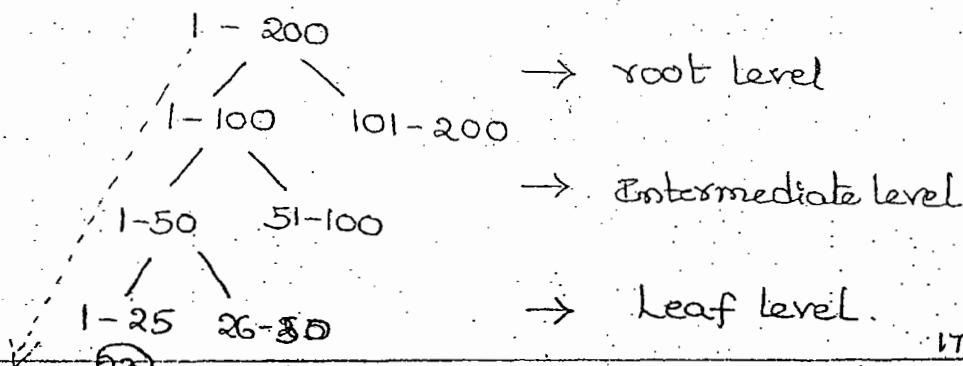
1. Clustered Index
2. Non clustered Index.

B-Tree Traversal Mechanism

All the data will organize in the form of Pages.

If we want to locate a data it has to scan all pages if we don't have indexes. When we use it will locate data very easily from Leaf Node and Root Node without performing more scans, it increases performance.

Ex :- If we want locate 23 page data in 200 pages using B Tree traversal Mechanism, Among total pages using mid value it searches that page record, further it goes to next level to locate data.



There is no need to scan all pages to locate data.

Among 200 pages 23rd page falls under 1-100 in Btree mid value, further it goes to 1-50 and leaves 51-100 in scanning then it comes to 1-25 Pages and locate 23rd page data fastly.

Clustered Index

What is the column inside a table most frequently used we configure clustered index on that column.

→ Clustered Index contains index value pointers that are same as which stored in data values.

→ All the values should be sorted in a proper order.

Clustered Index arranges these pages.

→ Leaf node of clustered index will have original data.

→ While creating it may take long time but when retrieving data it will be very fast.

Non Clustered Index

→ Non clustered Index cannot have same value of data values which stored in column. It is a pointer to that data column.

→ It points to clustered index for pointing data.

→ Leaf node of Nonclustered index will not have

Fragmentation

- When ever indexes are not performing upto mark we call it as fragmentation. when pages left empty not properly filled search will take more time here.
- We have 2 types of fragmentation.
 1. Internal fragmentation
 2. External fragmentation.

Internal fragmentation

out of 8kb page data, Pages left empty not filled properly inside. Due to this 4 pages data store in 8 pages. This is internal fragmentation.

External fragmentation

My data earlier store in 1 extent, due to shrinking operation (or) recompilation my required data goes to 10th extent instead of 1 extent. While searching it has scan all 10 extent pages. this is called External fragmentation.

To find we use

Select * from sys.dm_db_index_physical_stats.

This DMV will show all indexes information on server including system databases.

original data. It will get data indirectly by depending on clustered index. If clustered index is not there it will search by using file pointers of statistical structure

In 2005 we have 249 Non clustered and 1 clustered Indexes.

In 2008 the size increased in 1000, out of these 1 is clustered index and 999 are Non clustered.

Indexes can some time causes performance issues when we use over indexing, it will degrade performance. When we do not have clustered index we call that table as "Heek table". It will collect data from file pointers, page statistical information, page no and row no etc, it consumes time in retrieving data. We have one more type index called covering index.

Covering Index

When we building 2 or more indexes on columns of table, we can build a covering index on all these columns for sorting the data.

As a DBA if it is table scan we need to build index on that table. If it is a index scan it gives best performance always.

In DMV if we need on particular database we need to specify, By default Index Partition value is 1
There are 5 parameters required to run this DMV.
.Dbid, Objectid, Indexid, Index partition, mode.

We can pass NULL if we are not sure about parameters
output of this DMV →

Avg fragmentation in %

It shows level of fragmentation. It has to be always 0. If 0 → 5 we can ignore, More than 5 it's a problem.

5-20% we have Reorganize index

More than 20% we have to rebuild index.

Reorganize

All records will arrange properly inside the Index,

It will make data in order. It is online operation.

Alter Index All on Table name Reorganize

Used for indexes with a small fragmentation percentage

This is a online operation.

Rebuild

This is offline operation. It drops old indexes completely and create new indexes on table

We use this option when fragmentation percentage is more. We use Command.

ALTER INDEX index-name ON table-name REBUILD.

If we have more tables that we want to rebuild all of them, we use

ALTER INDEX ALL ON table-name REBUILD.

SQL Server 2012 Enhancements

In SQL Server 2012 we can perform online index rebuild for indexes. Earlier version we have certain limitations. When indexes contain large object datatype like XML, VARCHAR(MAX), IMAGE, TEXT we not able perform online index rebuild.

Now we can perform online index rebuilds for indexes containing these data types. This enhancement provides more flexible management capabilities in SQL Server.

QUERY TUNING

Query optimizer generates multiple plans in which way query need to execute. SQL server will choose base on cost I/O consumption, Response time, Memory consumption of a query.

Execution plan is generated once. It will store in plan cache of Buffer pool. In case we run same query again it directly comes to plan cache and provide results very quickly. Hence second time query execution will be faster. In case of New query it has to generate new execution plan. There is a standard procedure in executing queries if we use execution plan from plan cache it eliminates Parsing and recompilation of query.

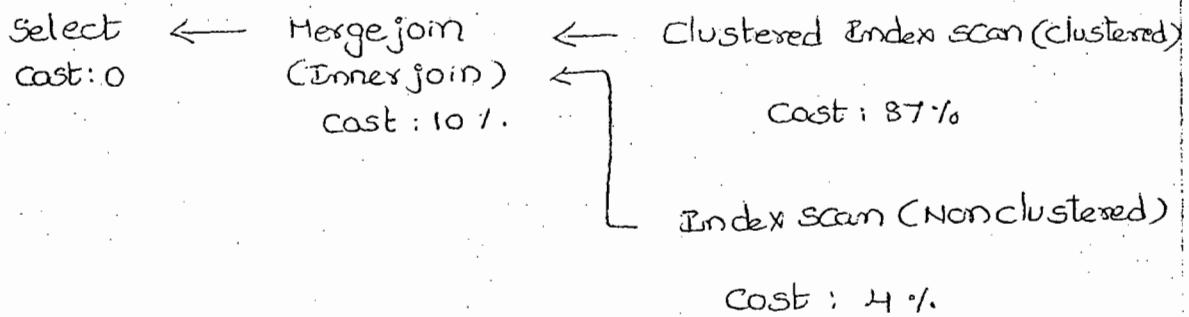
Execution Plans

Execution plan provide details steps how the query is executing. When ever queries taking long time to execute we need to tune the query by looking in to the execution plans.

Ex query: select * from msdb..backupset where type = 'd'
Press Ctrl+F5 → We can see Estimated plan of query.
Plan can be viewed from Right to Left.

We can get details what is index used, I/O cost of query, CPU cost, No of executions, operator cost,

row side we get all the details.



Type of scan

Table scan

Index scan

clustered Index scan

Index seek.

Table scan

Searching full table causes performance issues. Each and every row, column has to search. If we have Table scan in query, when there is no index and when we don't have covering index, Then Table scan occurs.

Index Scan

It uses indexes, Need to check why performance is high even though we have indexes on Table. It may be because of more scanning operations which may consumes high CPU process. Here we have tune

indexes

clustered index scan

We have 2 types of indexes (i) clustered Index

(ii) Non clustered Index.

Using clustered index we get always best performance

Even though if we have performance issues we will verify

Whether where clause and order by clause used in

the query (or) not need to verify. Where clause brings results directly without scanning the data.

Cluster index scan sort and stores data just like tables.

We recommend development Team not use excessive joins, joins keep more burden on sqlserver.

We can ask Keep the transaction logic in shorter batches.

Index seek

Index seek is a option getting what input we want.

Index seek work like a Bookend Index. It is going to locate using pointers directly bring which row we want.

Recommendations as a DBA for getting Good performance

1. We can ask development team to create transactions inside stored procedures. In a short means short batches.
Ex :- If we are deleting 1 million records, we ask them not delete all in a single transactions that impact performance. If use shorter batches like 10 lakh records in one batch will not impact performance.
2. choose an appropriate Escalation in transaction
3. Minimize the usage of joins in transactions.
4. Minimize usage of cursors.
5. Use Where and having clause in all queries.

Here we are forcing our queries to use index seek operations by using Where and having clause.