Files and Filegroups

Microsoft SQL Server

Rev 1.0-1406

By Naji El Kotob
naji [@] dotnetheroes.com
www.DotNETHeroes.com

Outlines

- Introduction to SQL Server files Architecture
  - Understanding Pages and Extents
  - About Files and Filegroups
  - Primary and Secondary Filegroups
  - Using Files and Filegroups
  - Read-Only Filegroups
  - Rules for Designing Files and Filegroups
  - Recommendations
  - RAID?
  - FileGroup and RAID
- Interactive in-class DEMO
- QnA
- References

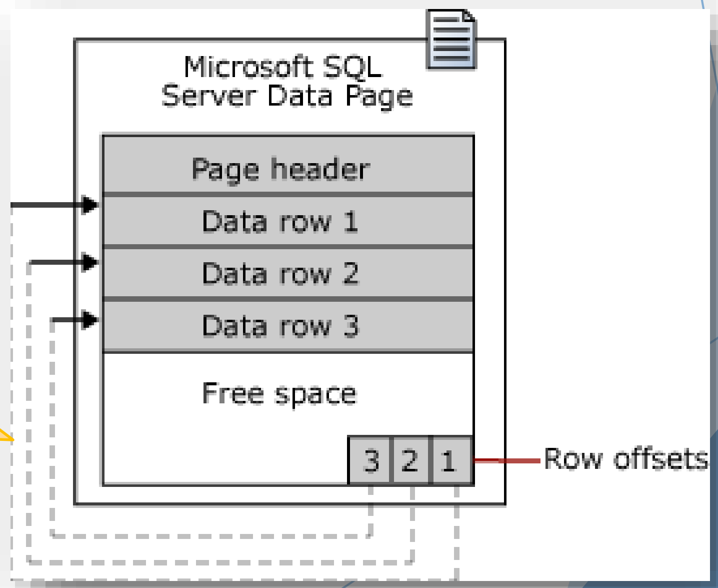
Database Sample

- AdventureWorks 2012 Database
  - *http://msftdbprodsamples.codeplex.com*

Understanding **Pages** and Extents

- Pages
  - The fundamental unit of data storage in SQL Server is the page.
  - The disk space allocated to a data file (.mdf or .ndf) in a database is logically divided into pages numbered contiguously from 0 to n.
  - Disk I/O operations are performed at the page level.
  - SQL Server reads or writes whole data pages.

In SQL Server, the page size is **8 KB**. This means SQL Server databases have **128 pages per megabyte**. Each page begins with a 96-byte header that is used to store system information about the page.

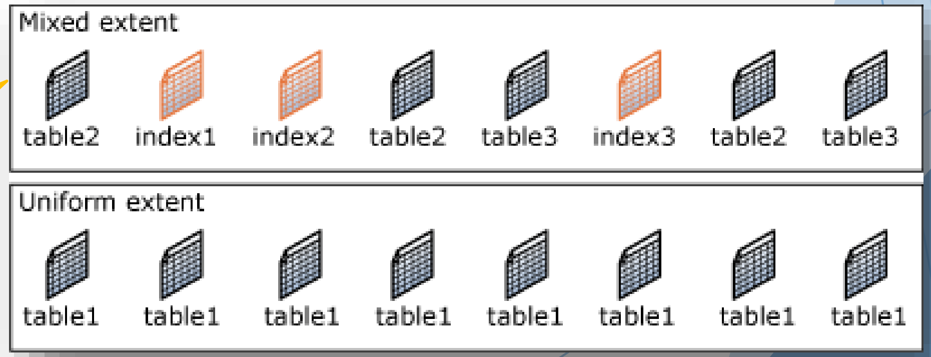


http://technet.microsoft.com/en-us/library/ms190969(v=sql.105).aspx

Understanding Pages and Extents

- Extents
  - Extents are a collection of eight physically contiguous pages and are used to efficiently manage the pages. All pages are stored in extents.
  - Extents are the basic unit in which space is managed.
  - An extent is eight physically contiguous pages, or 64 KB. This means SQL Server databases have 16 extents per megabyte.

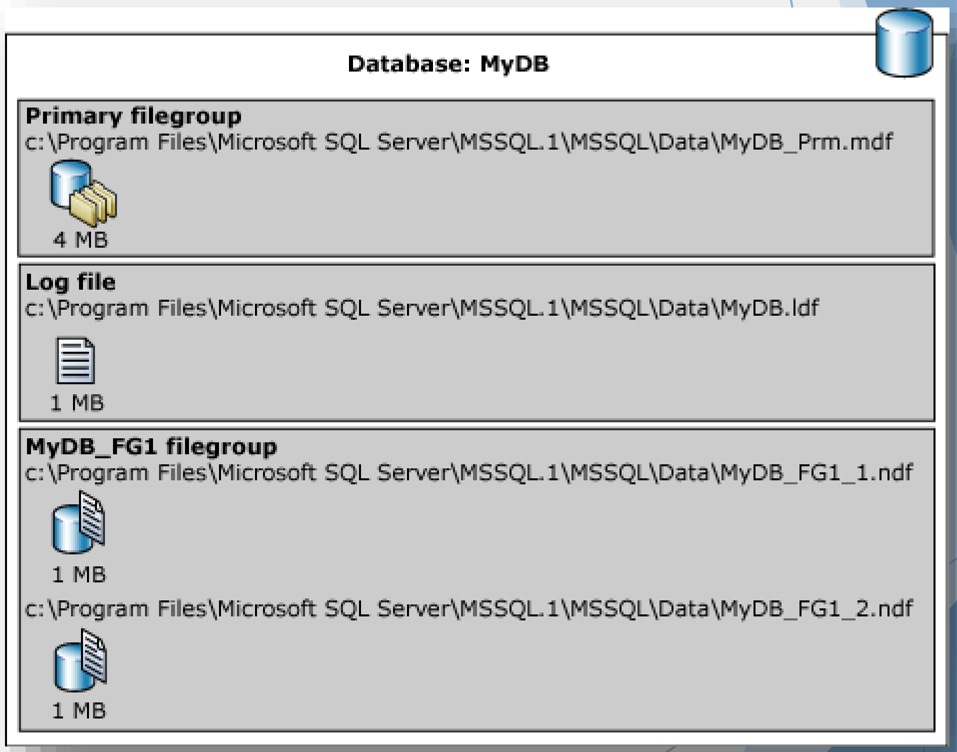
A new table or index is generally allocated pages from **mixed** extents. When the table or index grows to the point that it has eight pages, it then switches to use **uniform** extents for subsequent allocations



http://technet.microsoft.com/en-us/library/ms190969(v=sql.105).aspx

Files and Filegroups

- SQL Server maps a database over a set of operating-system files.
- Data and log information are never mixed in the same file, and individual files are used only by one database.
- **Filegroups** are named collections of files and are used to help with data placement and administrative tasks such as backup and restore operations.



Database: MyDB

Primary filegroup
c:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\Data\MyDB_Prm.mdf
4 MB

Log file
c:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\Data\MyDB.ldf
1 MB

MyDB_FG1 filegroup
c:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\Data\MyDB_FG1_1.ndf
1 MB
c:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\Data\MyDB_FG1_2.ndf
1 MB

http://technet.microsoft.com/en-us/library/ms179316(v=sql.105).aspx

Files default Extensions

- .MDF = Master Database File
- .LDF = Log Database File
- .NDF = Secondary Database File

Primary and Secondary Filegroups

- **A primary filegroup** contains the primary datafile (mdf).
  - Tip: All system tables are allocated to the primary filegroup.
- **A secondary filegroup** (also called a user-defined filegroup) contains secondary datafiles (ndf) and database objects.
- **The default filegroup** contains objects which were created without an assigned filegroup.
  - Tip: The primary filegroup is the default filegroup unless another filegroup is specified.
- **Logfiles** (.ldf) are never part of a filegroup.

Using Files and Filegroups

- File and Filegroup Fill Strategy
  - Filegroups use a proportional fill strategy across all the files within each filegroup.
    - For example, if file f1 has 100 MB free and file f2 has 200 MB free, one extent is allocated from file f1, two extents from file f2, and so on. In this way, both files become full at about the same time, and simple striping is achieved.
- Improving Database Performance
  - Using files and filegroups improves database performance, because it lets a database be created across multiple disks, multiple disk controllers, or RAID (redundant array of independent disks) systems.
- Implementing Backup and Restore Strategies
  - Databases made up of multiple filegroups can be restored in stages by a process known as piecemeal restore.

Read-Only Filegroups

- Any existing filegroup, except the primary filegroup, can be marked as read-only.
- A filegroup marked read-only cannot be modified in any way.
  - Tip: For tables that must not be modified, such as historical data, put them on filegroups and then mark the filegroup as read-only. This prevents accidental updates.

Rules for Designing Files and Filegroups

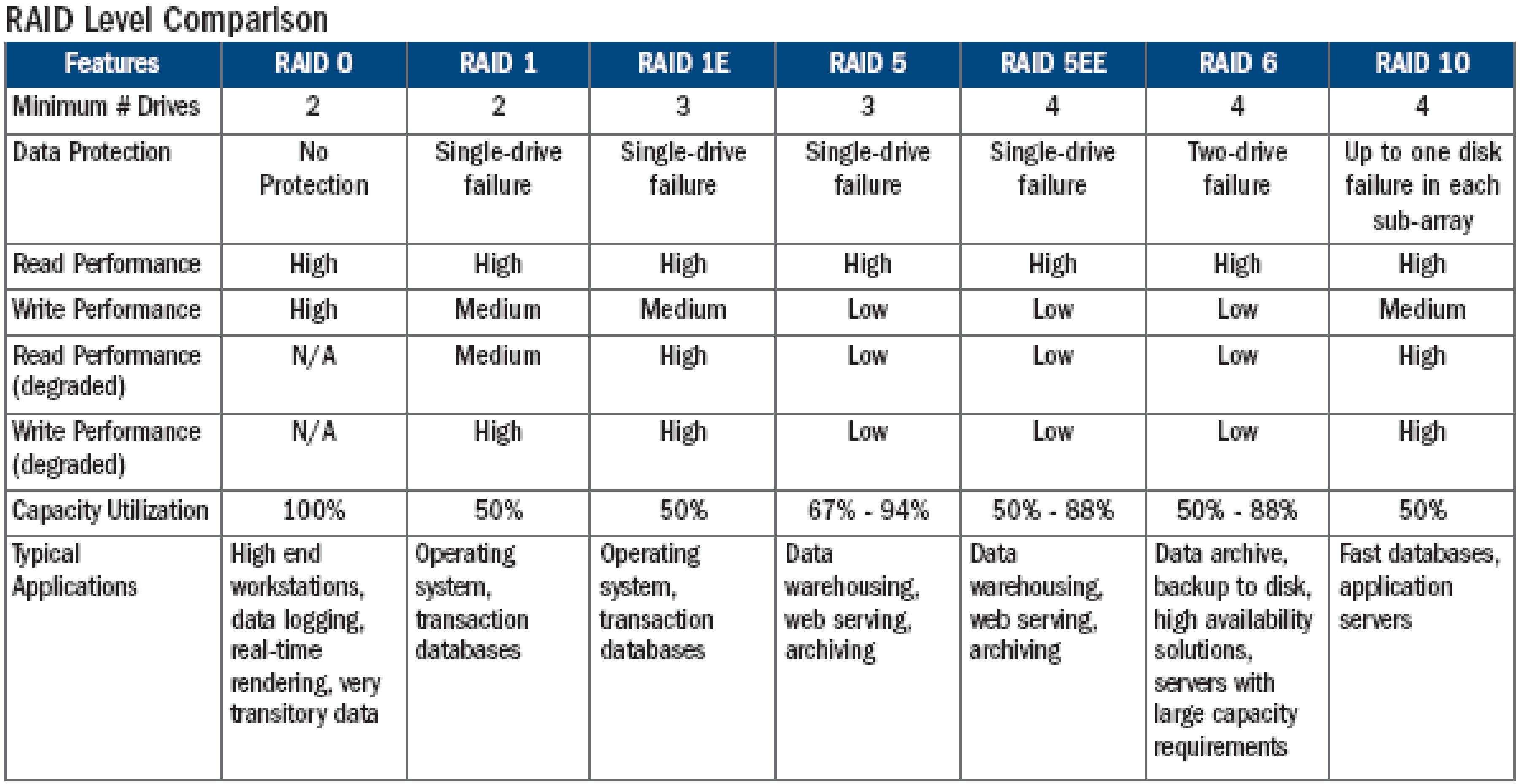
- The following rules pertain to files and filegroups:
- A file or filegroup cannot be used by more than one database. For example, file sales.mdf and sales.ndf, which contain data and objects from the sales database, cannot be used by any other database.
- A file can be a member of only one filegroup.
- Transaction log files are never part of any filegroups.

Files and Filegroups Recommendations

- Most databases will work well with a single data file and a single transaction log file.
- If you use multiple files, create a second filegroup for the additional file and make that filegroup the **default** filegroup.
  - Tip: In this way, the primary file will contain only system tables and objects.
- To maximize performance, create files or filegroups on as many different available local physical disks as possible.
  - Tip: Put objects that compete heavily for space in different filegroups.
- Use filegroups to enable placement of objects on specific physical disks.
- Put different tables used in the same join queries in different filegroups.
  - Tip: This will improve performance, because of parallel disk I/O searching for joined data.
- Put heavily accessed tables and the nonclustered indexes that belong to those tables on different filegroups.
  - Tip: This will improve performance, because of parallel I/O if the files are located on different physical disks.
- Do not put the transaction log file or files on the same physical disk that has the other files and filegroups.

For more info http://technet.microsoft.com/en-us/library/ms187087(v=sql.105).aspx

RAID



RAID Level Comparison

Features	RAID 0	RAID 1	RAID 1E	RAID 5	RAID 5EE	RAID 6	RAID 10
Minimum # Drives	2	2	3	3	4	4	4
Data Protection	No Protection	Single-drive failure	Single-drive failure	Single-drive failure	Single-drive failure	Two-drive failure	Up to one disk failure in each sub-array
Read Performance	High	High	High	High	High	High	High
Write Performance	High	Medium	Medium	Low	Low	Low	Medium
Read Performance (degraded)	N/A	Medium	High	Low	Low	Low	High
Write Performance (degraded)	N/A	High	High	Low	Low	Low	High
Capacity Utilization	100%	50%	50%	67% - 94%	50% - 88%	50% - 88%	50%
Typical Applications	High end workstations, data logging, real-time rendering, very transitory data	Operating system, transaction databases	Operating system, transaction databases	Data warehousing, web serving, archiving	Data warehousing, web serving, archiving	Data archive, backup to disk, high availability solutions, servers with large capacity requirements	Fast databases, application servers

http://www.gatan.com/images/knowhow/kh15_im3.png

FileGroup VS. RAID

- The best performance of your SQL Server database is to combine both multiple Files/Filegroups and hardware RAID together.

- E.g.
  - 4 disks in a RAID-10 array for the data.
  - 2 disks using RAID-1 (mirroring) for the transaction log.
  - 1 disk for the non-clustered indexes

Q&A

- Please send your feedback to naji [@] dotnetheroes.com

References

- http://technet.microsoft.com/en-us/library/ms179316(v=sql.105).aspx (File and Filegroups Architecture)
- http://www.prepressure.com/library/technology/raid
- http://macperformanceguide.com/Storage-RAID1-speed.html