Reduce SQL Server Index Fragmentation

When you perform any data modification operations (INSERT, UPDATE, or DELETE statements) table fragmentation can occur. When changes are made to the data that affect the index, index fragmentation can occur and the information in the index can get scattered in the database. Fragmented data can cause SQL Server to perform unnecessary data reads, so a queries performance against a heavy fragmented table can be very poor.
The image below is a graphical presentation of how SQL Server lays out the data.
Use the following SQL script to determine the fragmentation state of tables within a database (Any SQL Server Version):

```
 USE [Database]
GO
EXEC sp_MSforeachtable @command1="print '?' DBCC SHOWCONTIG('?')"
GO
```

OR, to list for each table in each database (SQL Server 2005+ only):

```
use master
GO
EXEC sp_msforeachdb '
  BEGIN
   IF NOT(("$" = "master") OR ("$" = "tempdb") OR ("$" = "model") OR ("$" = "msdb"))
   BEGIN
    PRINT "----------------------------------------------"
    PRINT "$"
    PRINT "----------------------------------------------"
    EXEC [$].dbo.sp_msforeachtable @command1="print ""?"" DBCC SHOWCONTIG(""?"")
     ", @replacechar="?"
   END
  END
', @replacechar='$'
```

The output will look something like the example below, which was run on the SlyStock database:

```
[dbo].[Adjustment]
DBCC SHOWCONTIG scanning 'Adjustment' table...
Table: 'Adjustment' (2121058592); index ID: 1, database ID: 11
TABLE level scan performed.
- Pages Scanned.....: 11044
- Extents Scanned.....: 3804
- Extent Switches.....: 10866
- Avg. Pages per Extent.....: 2.9
- Scan Density [Best Count:Actual Count].....: 12.71% [1381:10867]
- Logical Scan Fragmentation .....: 48.14%
- Extent Scan Fragmentation .....: 74.50%
- Avg. Bytes Free per Page.....: 3129.6
- Avg. Page Density (full).....: 61.33%
DBCC execution completed. If DBCC printed error messages, contact your system administrator.
```

In the example above we see that there were 11,044 pages examined to create the report. Those pages existed within 3,804 extents. We then see that while examining the pages for fragmentation, the server had to switch extent locations 10,866 times. The Scan Density indicates the percentage of all extents where the pages were contiguous. In an ideal environment, the density displayed would be close to 100.

The Logical Scan Fragmentation and Extent Scan Fragmentation are indications of how well the indexes are stored within the system when a clustered index is present (and should be ignored for tables that do not have a clustered index). In both cases, a number close to 0 is preferable. There is another anomaly being displayed here; SQL Server allows multiple tables to exist within a single extent, however multiple tables may not exist within a page.

The Avg. Bytes Free per Page shows that there are an average of 3130 (3129.6) bytes free per page, or that each page is 61.33% utilized. The closer that number gets to 100, the faster the database is able to read in records, since more records exist on a single page. However, this must be balanced with the cost of writing to the table. Since a page split will occur if a write is required on a page that is full, the overhead can be tremendous. This is exaggerated when using RAID 5 disk subsystems, since RAID 5 has a considerably slower write time compared to its read time. To account for this, we have the ability of telling SQL Server to leave each page a certain percentage full.

In Summary, the Scan Density should be as close to 100 as possible, both the Logical Scan Fragmentation and Extent Scan Fragmentation should be as close to 0 as possible, and the Page Density should be around 65-75%.

For more information please visit Microsoft SQL Server Index Defragmentation Best Practices.

You can reduce fragmentation and improve read-ahead performance by using one of the following:

Dropping and re-creating an index. Rebuilding an index by using the DBCC DBREINDEX statement. Defragmenting an index by using the DBCC INDEXDEFRAG statement.

http://www.wintill.com/kb/questions.php?questionid=39