SQL SERVER Performance Tuning Notes

http://msbiskills.com/

LIKE IS SARGable, How to write better Where Clause –II. Let's Check?

SQL Server has many functions that can be used in your WHERE clause and in the SELECT clause. Now these functions can be user defined or inbuilt. These functions normally provide functionality which would be very difficult to get without these functions.

Now when these functions are used improperly in the WHERE clause these functions can cause major performance issues. Now very few SQL functions are SARGable and most of them are NOT SARGable.

Now what is SARgable?

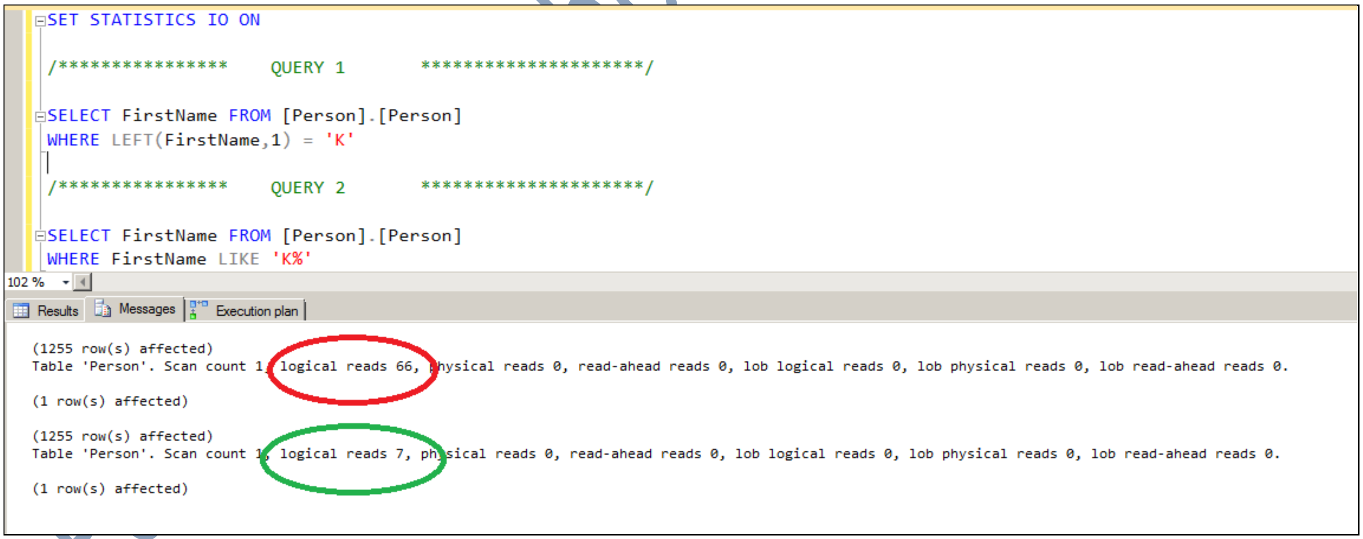
It is Search Argument able. It's the ability of the query optimizer to use indexes.

Let's see an example that shows using a function in the WHERE clause can affect performance. Add actual execution plan and statistics IO ON for better understanding.

Connect to AdventureWorks2012 & create a NonClustered Index on First Name. Now we are finding all the persons having name start with Letter "K". Let's check two queries to get desired results and their execution plan.

```
SET STATISTICS IO ON

/****************    QUERY 1       *********************/


SELECT FirstName FROM [Person].[Person]
WHERE LEFT(FirstName,1) = 'K'


/****************    QUERY 2       *********************/


SELECT FirstName FROM [Person].[Person]
WHERE FirstName LIKE 'K%'
```

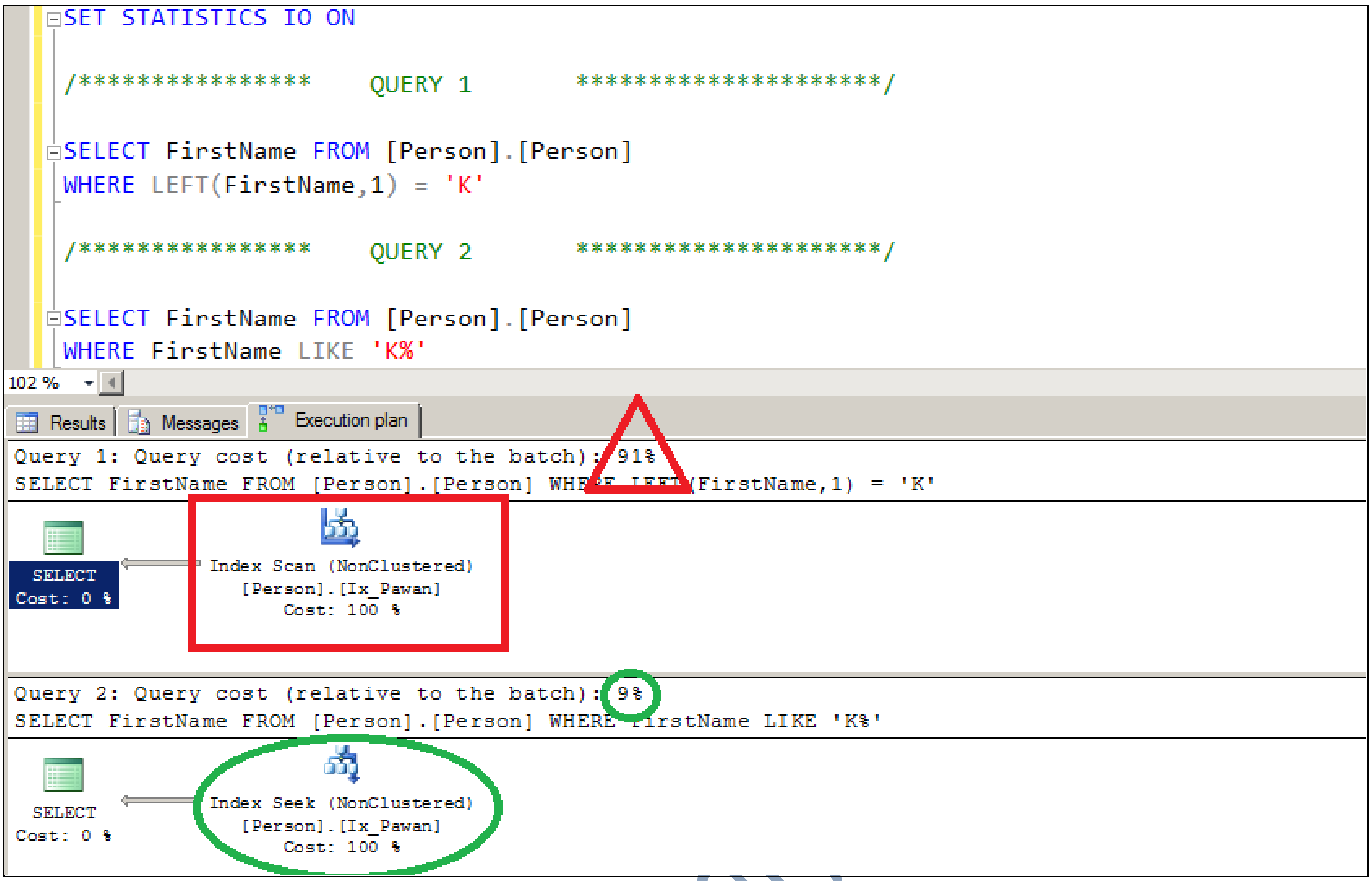
First let's check out the messages tab. Both the above queries are working fine and we got 1255 rows in both the case. Now let's check the logical reads (1 Logical Read = 8KB Pages) for both the queries. First query is reading 66 pages and the second query is reading only 7 pages. It is clearly evident that second query is performing much better than the first one.



```
SET STATISTICS IO ON

/****************    QUERY 1       *********************/

SELECT FirstName FROM [Person].[Person]
WHERE LEFT(FirstName,1) = 'K'

/****************    QUERY 2       *********************/

SELECT FirstName FROM [Person].[Person]
WHERE FirstName LIKE 'K%'
```

```
(1255 row(s) affected)
Table 'Person'. Scan count 1, logical reads 66, physical reads 0, read-ahead reads 0, lob logical reads 0, lob physical reads 0, lob read-ahead reads 0.

(1 row(s) affected)

(1255 row(s) affected)
Table 'Person'. Scan count 1, logical reads 7, physical reads 0, read-ahead reads 0, lob logical reads 0, lob physical reads 0, lob read-ahead reads 0.

(1 row(s) affected)
```

Now let's look out the execution plan for both the queries.



```
SET STATISTICS IO ON

/****************     QUERY 1      *********************/

SELECT FirstName FROM [Person].[Person]
WHERE LEFT(FirstName,1) = 'K'

/****************     QUERY 2      *********************/

SELECT FirstName FROM [Person].[Person]
WHERE FirstName LIKE 'K%'
```

```
Query 1: Query cost (relative to the batch): 91%
SELECT FirstName FROM [Person].[Person] WHERE LEFT(FirstName,1) = 'K'

SELECT          Index Scan (NonClustered)
Cost: 0 %       [Person].[Ix_Pawan]
                Cost: 100 %

Query 2: Query cost (relative to the batch): 9%
SELECT FirstName FROM [Person].[Person] WHERE FirstName LIKE 'K%'

SELECT          Index Seek (NonClustered)
Cost: 0 %       [Person].[Ix_Pawan]
                Cost: 100 %
```

We can clearly check out the Query 1 is having Index Scan means reading all the records of the table from leaf level and it is taking 91% of the cost. On the other hand second query is having nonclustered seek and taking only 9% of the cost. Now if we use second query then our system can become at least 10% more scalable; which is a very good thing.

Summary

When we are checking things for performance always try to find out these small things. These can help you queries performance greatly. We are reducing I/O by reading less data pages. So by not using functions in the WHERE clause we can provide big performance gains. We just have to tweak where clause. That should be easy. ☺

That's all folks; I hope you've enjoyed learning about Like IS SARGable, How to write better Where Clause –II and I'll see you soon with more "Performance Tuning" articles.

Thanks!

Pawan Kumar Khowal

MSBISKills.com