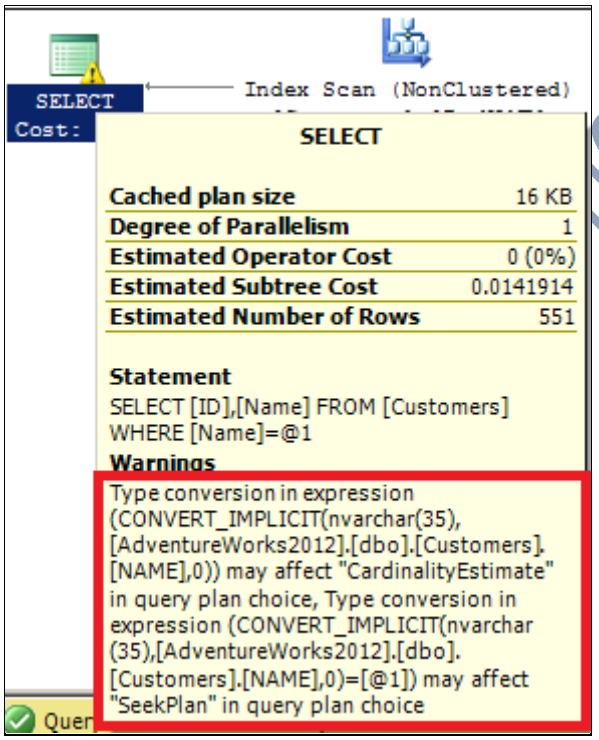
SQL SERVER Performance Tuning Notes

http://msbiskills.com/

Implicit Conversions are Bad for Performances? Why?

What is Implicit Conversion?

When you write your T-SQL queries, you often write where clause and compare column values with expressions and constants. Now if these things are NOT of same data type then SQL Server will NOT throw any error for it but internally it will implicitly convert one of the data type to match with the other one. This is called Implicit Conversion. Common places where we see implicit conversions are where and from clause.



Pic - Implicit Conversion

You can see the Implicit Conversion in execution plan by hovering over the warning over operators (Like we have on Select Operator). Well Implicit

conversion is NOT a good thing in the execution plan, and can lead to excessive CPU use.

In some cases, converting from one type to another may cause a loss of precision also. Please check out the chart showing all the data types and their conversions at http://msdn.microsoft.com/en-us/library/ms187928.aspx.

Now the million dollars question why Implicit Conversion is bad?

It is bad because in this case we will get a sub optimal plan. Let's go through an example of Implicit Conversion in a where clause. Before that let's first create a table called Customers, insert some data and create some indexes on it. In this example we are comparing Varchar value to an NVarchar value.

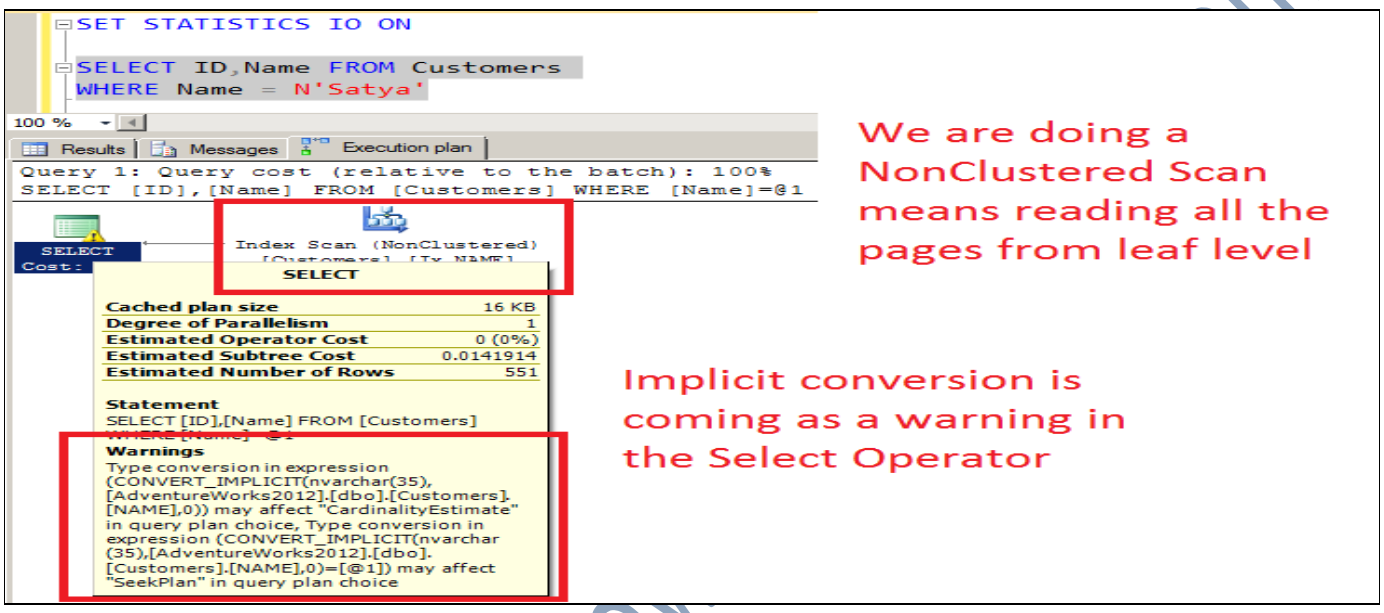
```
--Create a Sample table

CREATE TABLE Customers
(
       ID INT
      ,NAME VARCHAR(35)
)
GO

--Insert couple of data

INSERT INTO Customers VALUES
(1,'Ramesh'),(2,'Ganesh'),(3,'Isha'),(4,'Sharlee'),(5,'Satya'),(6,'Rajesh')


--Insert some more rows using a number table

INSERT INTO Customers
SELECT t.number , 'Rims'
      FROM master..spt_values t
CROSS APPLY
   (SELECT Number FROM master..spt_values q WHERE q.number = t.number ) r
WHERE t.Number between 1 and 5000
```

Create some indexes on the customers table.

```
CREATE CLUSTERED INDEX Ix_ID ON Customers(ID)
CREATE NONCLUSTERED INDEX Ix_Name ON Customers(Name)
```

Now we execute below query and check out its execution plan.

```
SELECT ID,Name FROM Customers WHERE Name = N'Satya'
```



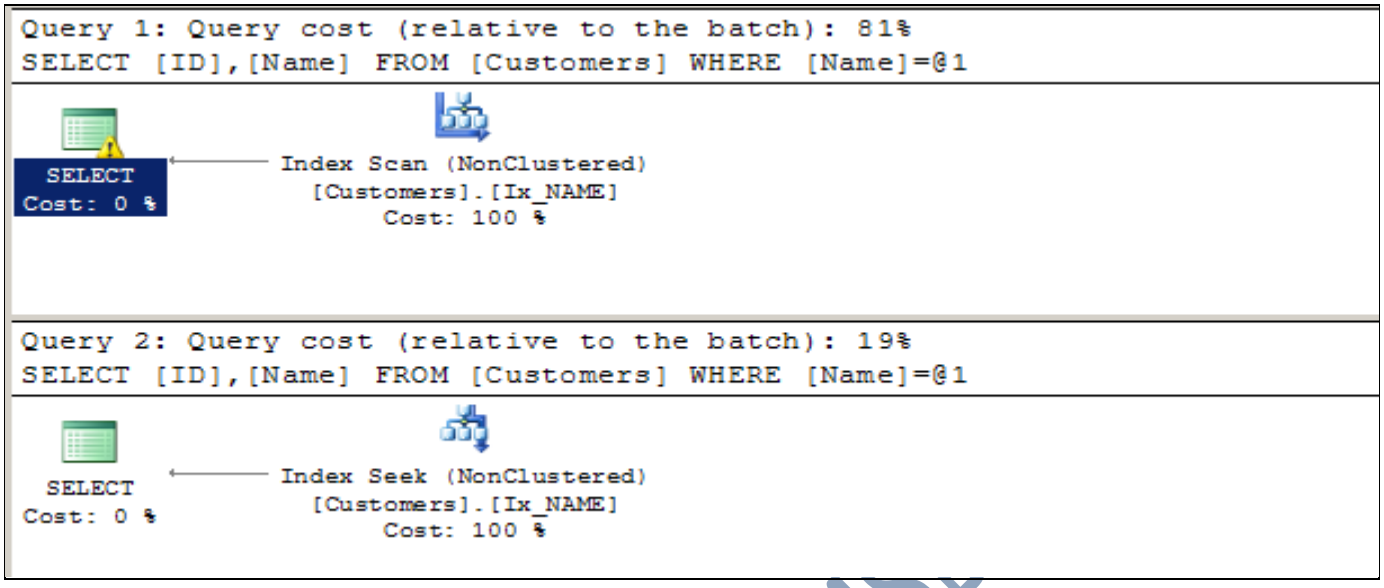
In the above query we are reading complete leaf level pages using nonclustered index scan even if we have a nonclustered index on Name column and on top of that we are getting an implicit conversion so all in all we can say that our query is not performing well.

Now the question is why the optimizer is converting Name from Varchar data type to Nvarchar data type. This is because of data Type Precedence. When an operator combines two expressions of different data types, the rules for data type precedence specify that the data type with the lower precedence is converted to the data type with the higher precedence.

Now let's rewrite our query.

```
SELECT ID, Name FROM Customers WHERE Name = 'Satya'
```

Please note that we have just removed the character **N** from our old query. Now let's execute both the queries and check their execution plans.



Ok so our old query is taking 81% cost and the second query is taking 19% of the cost and we have an index seek in 2nd query compared to index scan in the first query. So it is clearly evident that due to implicit conversion the cost of our query grows from 19% to 81% and also notes that we are reading complete table which is not necessary since in our case we have a nonclustered index on LastName column. All in this entire means our query is 4 times more scalable.

Summary

Now if you have a proper database design in place and if you prevent implicit conversions then your queries will perform better as less CPU and IO usage will be there. Basically you got to understand how your query optimizer works and sometimes if you can help him a little bit then you application can perform in an excellent manner. Now even you have a do a conversion do it explicitly so that SQL Server don't have to perform it internally for you.

That's all folks; I hope you've enjoyed learning about how we can eliminate implicit conversions for performance, and I'll see you soon with more "Performance Tuning" articles.

Thanks!

Pawan Kumar Khowal

MSBISKills.com