

# SQL SERVER Performance Tuning Notes

<http://msbiskills.com/>

## Execution/Query Plan Operator – The Sort Operator - Part II, Which sorting algorithms used by SQL Server? Is it blocking or non-blocking?

### Notes only

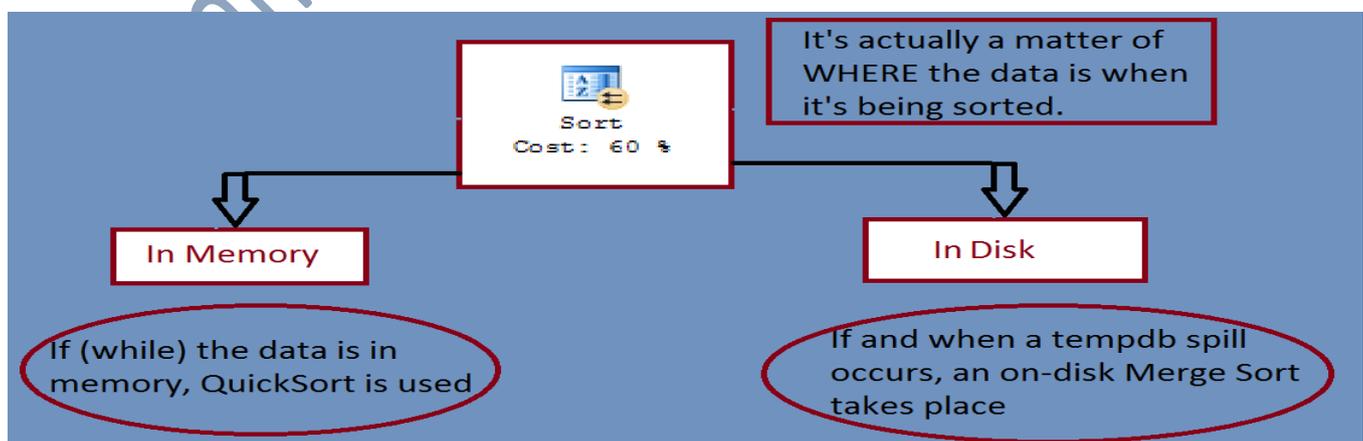
In this post I would like to talk sorting algorithms SQL Server uses for sorting. Well SQL Server uses 2 algorithms (or their flavours) to sort data. They are

- **Quick Sort**
- **Merge Sort**

It's actually a matter of WHERE the data is when it's being sorted. If & while the data is in memory, Quicksort is used.

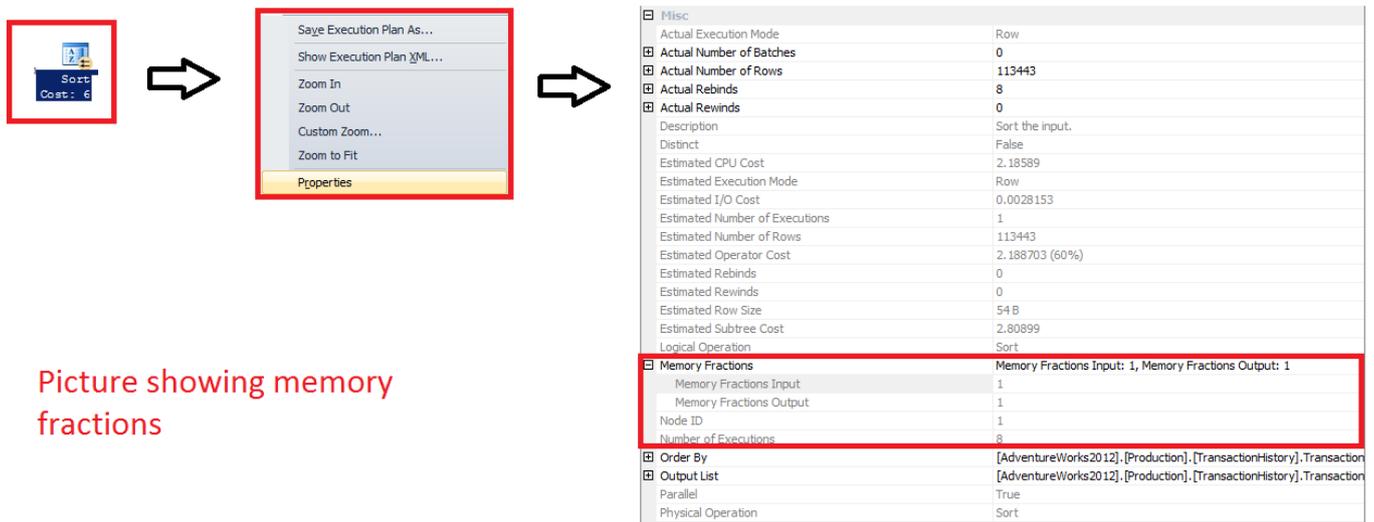
If and when a tempDB spill occurs, an on-disk Merge Sort takes place.

SQL Server begins sorting in memory using Quick Sort algorithm. Memory grant required in this case =  $200\% * \text{Input Size}$ . If the memory grant is increased it will spill everything (Entire immediate sort and remaining input rows). After that the sorting will be completed on disk using Merge Sort algorithm.



Picture – Sorting Algorithms

Now what is Memory grant fraction – It is a number between 0 and 1. 1 means 100% of the granted memory. One can view this by right clicking the sort operator and looking in the properties window. The example below was taken from a query with only a single Sort operator, so it has the full query workspace memory grant available during both input and output phases:



Picture - showing Memory grant fraction-

SQL Server has two kinds of execution plan operators. They are

- Blocking
- Non-blocking

For details about blocking and non-blocking operators please click here –

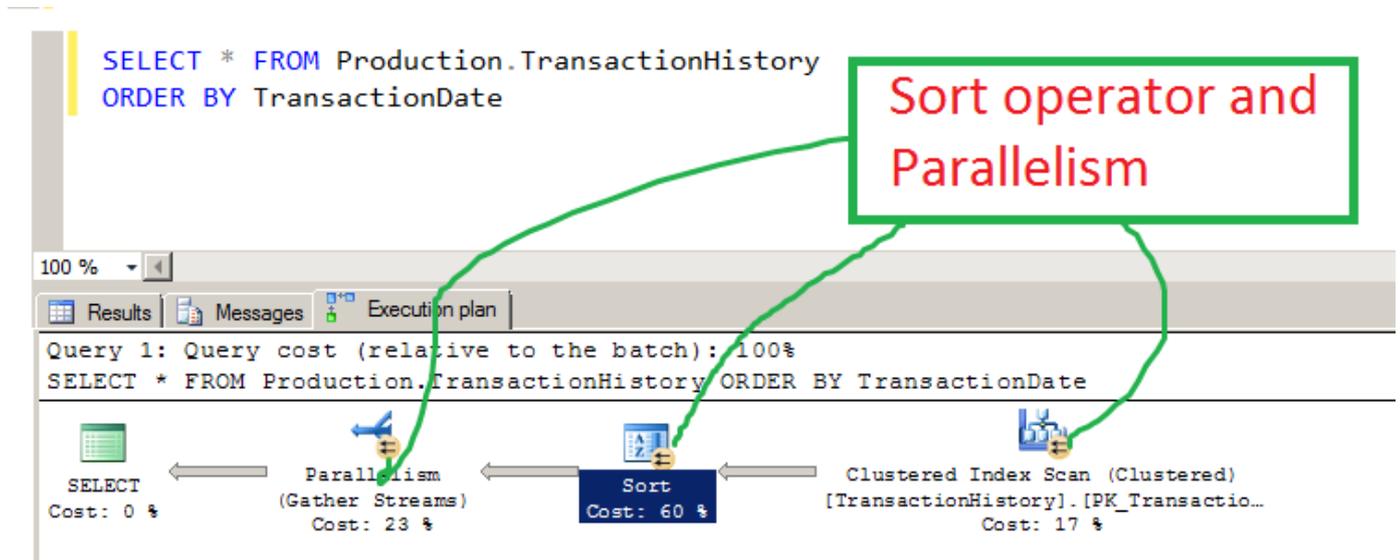
<http://msbiskills.com/2015/07/05/execution-plan-operator-properties-types-blocking-non-blocking/>

Now we know that Sort is a blocking operator, as it is very clear that unless we have all the data how can we sort as we don't know which row will be on the top and which one at the bottom. Now let's see the below query and its execution plan.

The Query is

USE AdventureWorks2012  
GO

```
SELECT * FROM Production.TransactionHistory
ORDER BY TransactionDate
```

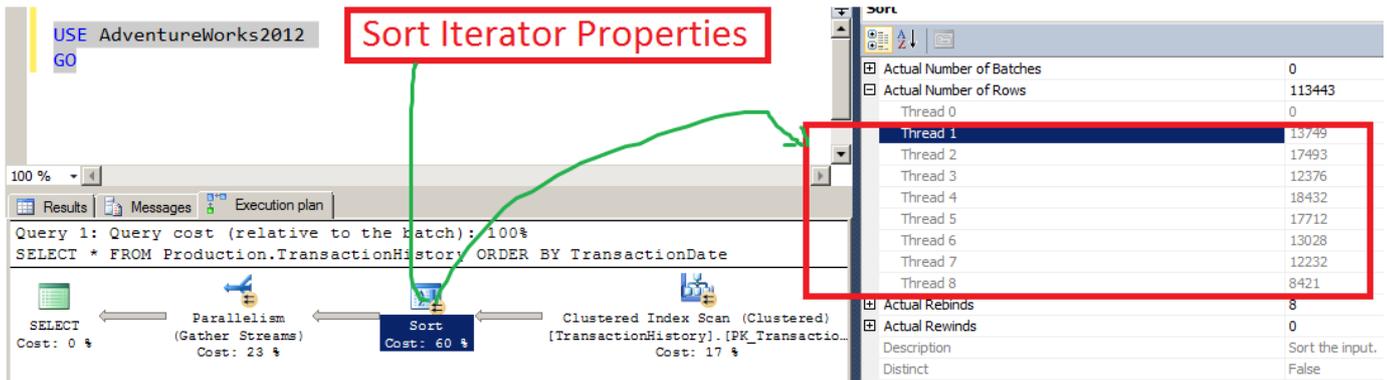


Picture showing execution plan - The sort operator and the parallelism.

This is a parallel plan; all the operators are working in parallel (Parallelism – is a vast topic, shall cover this basics about this in upcoming posts) but the root iterator. Now we know that sort is a blocking operator so why we have parallel symbol inside it. That is because Sort iterator has to consume all the input rows (In Open() Method) before it starts sorting but it starts producing output rows once the final run starts.

Now let's check the properties of sort operator and different threads to verify actually whether sort happens in parallel or not? Here the data on each thread is sorted separately and then Gather streams takes all the inputs and combine then together into a single stream.

There are eight threads present in the sort iterator properties because my laptop is an 8 core laptop. Basically numbers of threads are determined by MAXDOP (Maximum degree of Parallelism) for the plan.



Picture showing sort operator properties – Threads.

## Summary

1. Sort is a blocking operator. It starts producing output rows once the final run starts.
2. Sort is not a parallel aware iterator
3. It uses Quick Sort and Merge Sort (or their Variants internally) to sort data.

That's all folks; I hope you've enjoyed learning about Sort operator and its details, and I'll see you soon with more "Performance Tuning" articles.

Thanks!

Pawan Kumar Khawal

[MSBISkills.com](http://MSBISkills.com)