

SQLServerFast.com

Execution Plan Video Training

Block 1: Understanding execution plans

Level: Advanced

Chapter 2: Common operator properties

Common operator properties

Estimated Number of Rows

Estimated Number of Rows Per Execution

Estimated Number of Rows for All Executions

Actual Number of Rows

Actual Number of Rows for All Executions

*All these properties are explained
in block 1, basic level, chapter 6*

Estimated Number of Executions

Number of Executions

Common operator properties

Output List

*For basic functionality of operators,
see block 1, basic level, chapter 3*

GetNext() method makes operator do work

Results of GetNext(): one row of data returned to the caller

Columns in that row are specified in *Output List* property

Used by execution engine to ensure proper results

Can be used by us to better understand the execution plan

Track where data originates and where it is used

Query Optimizer – plan choice

Cardinality estimator

- Estimate amount of rows / executions for each operator

- Compute *Estimated I/O Cost* and *Estimated CPU Cost*

 - Per execution, based on cardinality estimates and operator specifics

- Compute *Estimated Operator Cost*

 - For all executions

- Compute *Estimated Subtree Cost*

 - Equal to total plan cost on left-most operator

Query Optimizer – plan choice

Estimated Subtree Cost / plan cost

“Seconds on Nick’s machine”



Source: <https://sqlstudies.com/2017/04/17/what-is-the-cost-in-cost-threshold-for-parallelism/>

Query Optimizer – plan choice

Estimated Subtree Cost / plan cost

“Seconds on Nick’s machine”

“Just a number” – useful to compare plans but for nothing else

Computed for **every** plan the optimizer considers

Compared to other possible plans to find the “cheapest” plan

Not guaranteed to actually be the fastest option

- Estimates may have been wrong

- Costing model based on outdated hardware assumptions

- May result in wrong priorities

Query Optimizer – plan choice

Estimated Subtree Cost / plan cost

“Seconds on Nick’s machine”

“Just a number” – useful to compare plans but for nothing else

Useful to help understand plan choices

Force plan you think would be better

Total plan cost shows why it was not chosen

Estimated Operator Cost on operators helps pinpoint problem area

Remove plan forcing after root cause is found and fixed

Execution plan plus run-time statistics

Actual I/O Cost?

Actual CPU Cost?

Actual Operator Cost?

Actual Subtree Cost?

Execution plan plus run-time statistics

~~Actual I/O Cost~~

~~Actual CPU Cost~~

~~Actual Operator Cost~~

~~Actual Subtree Cost~~

Only useful for plan choice

Not relevant* during / after execution

* (According to Microsoft)

Execution plan plus run-time statistics

Other “execution plan plus” properties

For execution plan as a whole

QueryTimeStats

WaitStats

*These were explained in
block 1, advanced level, chapter 1*

For individual operators

Actual I/O Statistics

Equivalent of SET STATISTICS IO, but tracked per operator

Actual Time Statistics

Not documented, and in many cases unclear

Better use *QueryTimeStats* on top left operator

Common operator properties

Logical Operation

Nested Loops	
For each row in the top (outer) input, scan the bottom (inner) input, and output matching rows.	
Physical Operation	Nested Loops
Logical Operation	Inner Join
Estimated Execution Mode	Row
Estimated I/O Cost	0
Estimated Operator Cost	0,507087 (5%)
Estimated CPU Cost	0

Nested Loops
(Inner Join)

Nested Loops	
For each row in the top (outer) input, scan the bottom (inner) input, and output matching rows.	
Physical Operation	Nested Loops
Logical Operation	Left Semi Join
Estimated Execution Mode	Row
Estimated I/O Cost	0
Estimated Operator Cost	0,507087 (5%)
Estimated CPU Cost	0

Nested Loops
(Left Semi Join)

Common operator properties

Tracking data through the execution plan

- Output List* to see what column(s) an operator returns

- Defined Values* to see how new columns are computed

 - Specifies, for each “new” column, how it is computed

Use these two together to understand the plan logic

Query tuning

The easy way is not always the best way

It may appear to help

But is it optimal?

Sometimes the key is not in the execution plan at all

Sometimes it is

And sometimes, what appears to be more efficient is actually not!

There is no easy way

(Sorry)

Summary

“Execution Plan Reference”:
sqlserverfast.com/epr

Common operator properties

- Determine *exact* actions of the operator

- Show how data travels through the execution plan

- Report on amount of work estimated to be done and actually done

- Insight in costs → insight in optimizer choices

“Obvious” optimizations are not always as simple as they look
Better understanding of execution plan helps find better optimizations

Next chapters

Chapter 3: Ordering of the data stream

- Operators that require ordered input

- Operators that set, retain, or disturb order of data stream

- Order-related optimizations

Chapter 4: Missing nodes

Chapter 5: Batch mode versus row mode