INTELIGENTNO PROCESIRANJE U SQL SERVERU 2019

MILOŠ RADIVOJEVIĆ

DATA PLATFORM MVP

26. DECEMBAR 2020

INTERLEAVED EXECUTION

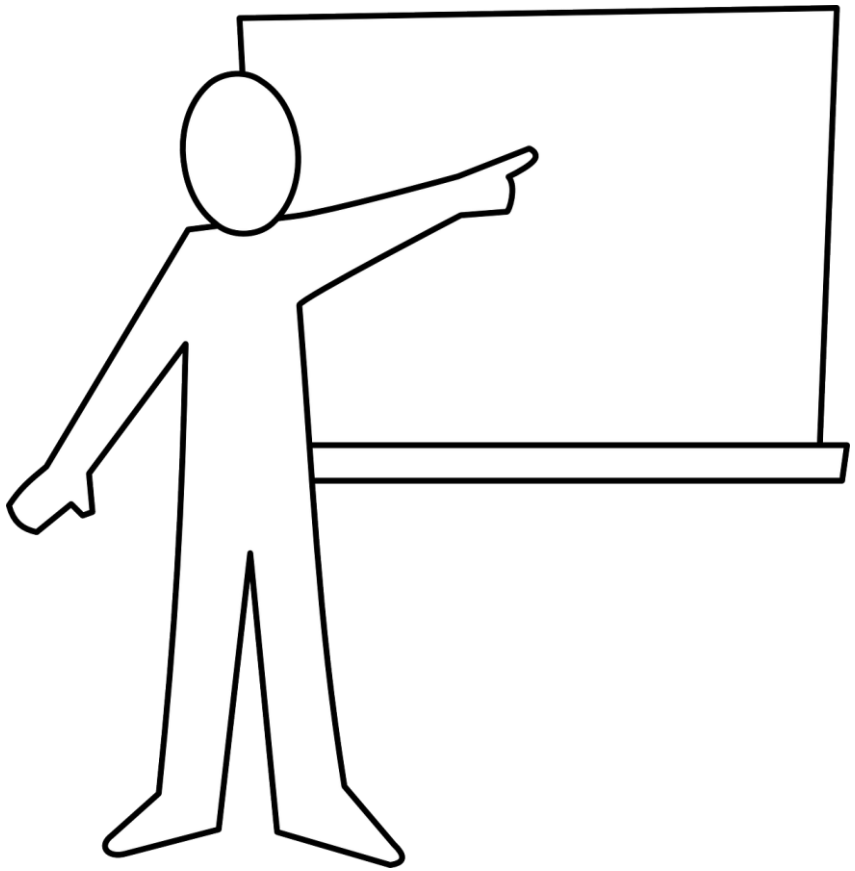
INTERLEAVED EXECUTION

- Improvement of the execution plan during generation
- Use the multi-statement table-valued functions (MSTVFs) for queries only and only part of it!
- On the other hand, the feature is revolutionary

USER DEFINED FUNCTIONS IN SQL SERVER

- Scalar functions
- Inline table-valued functions 
- Multi-statement table-valued functions (MSTVF)

PROBLEMS WITH MSTVFS

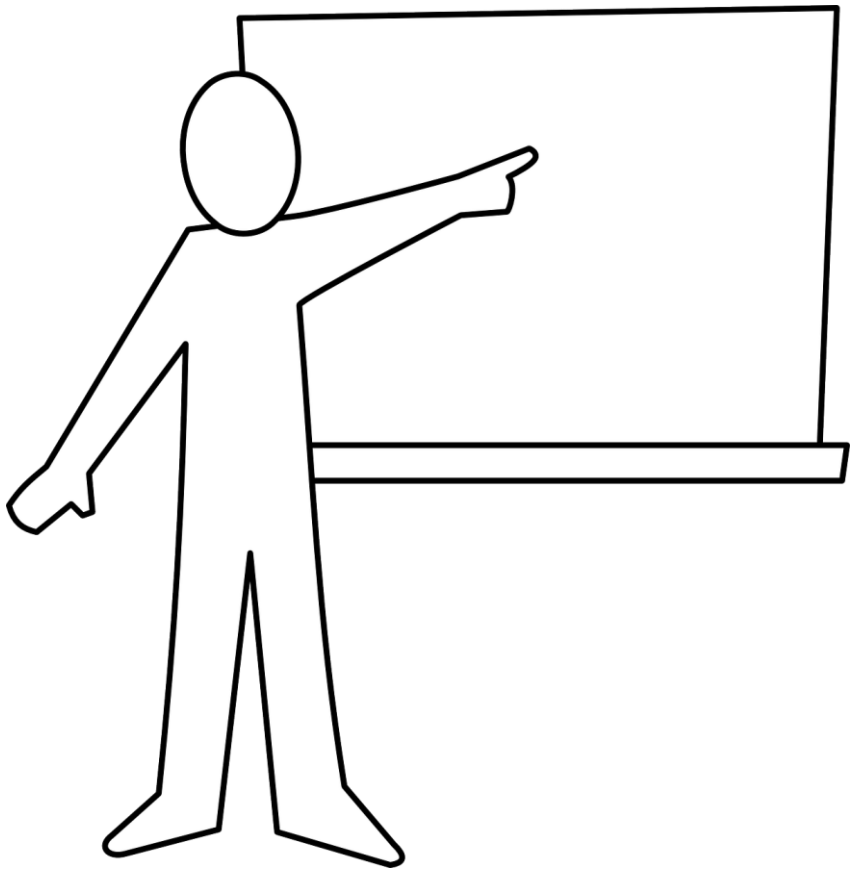


DEMO

ABFRAGEN MIT MSTVFS

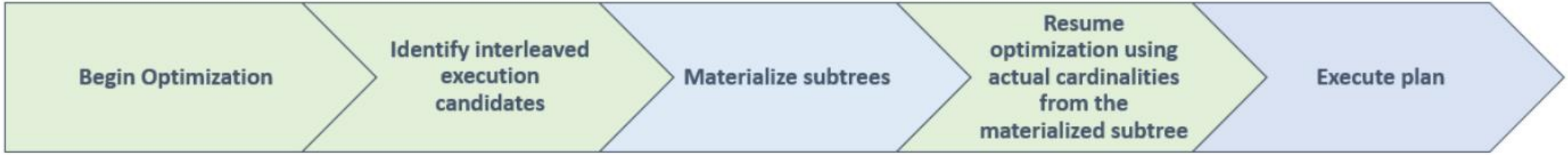
- Vor SQL Server 2017
  - geschätzte Anzahl der Zeilen zu niedrig
    - => Ausführungsplan mit dem Nested Loops Join Operator
    - => zugewiesene Speicher unzureichend
    - => tempdb spills
- MSTVFs mit der verschalteten Ausführung
  - geschätzte Anzahl der Zeilen = tatsächliche Anzahl der Zeilen
    - => Optimierer kann den geeigneten Operator auswählen
    - => zugewiesene Speicher
    - => no tempdb spills
- Im geschätzten Ausführungsplan, ist die MSTVF-Kardinalität immer noch 100

INTERLEAVED EXECUTION



DEMO

HOW IT WORKS?



- If an MSTVF is called, the optimization is interrupted
-
- MSTVF is executed and the result materialized
- The optimization process continues with the cardinality of the materialized rowset
- Execution Engine does not need to execute the function again

HOW IT WORKS?

```
SELECT ol.OrderID, ol.UnitPrice, ol.StockItemID
FROM Sales.Orderlines ol
INNER JOIN dbo.SignificantOrders() f1 ON f1.Id = ol.OrderID WHERE PackageTypeID = 7;
```

```
DROP TABLE IF EXISTS #T;
SELECT Id INTO #T FROM dbo.SignificantOrders();

SELECT ol.OrderID, ol.UnitPrice, ol.StockItemID
FROM Sales.Orderlines ol
INNER JOIN #T f1 ON f1.Id = ol.OrderID WHERE PackageTypeID = 7;
```

- But in the binding phase of compilation

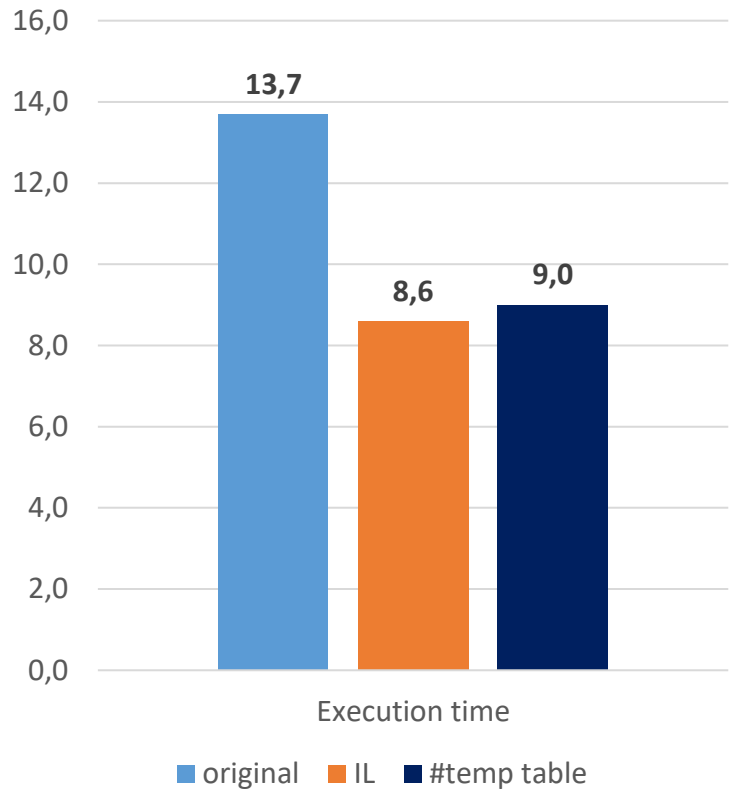
HOW GOOD IS INTERLEAVED EXECUTION?

- Query 1: original (CL 130)
- Query 2: with Interleaved Execution (CL 140)
- Query 3: Workaround with a temp table

```
C:\github\IT-Tage\Workshop\IL\ostress>cmd1

12/04/20 23:05:59.819 [0x00025768] Total IO waits: 0, Total IO wait time: 0 (ms)
12/04/20 23:05:59.819 [0x00025768] OSTRESS exiting normally, elapsed time: 00:00:13.696
12/04/20 23:05:59.820 [0x00025768] RsFx I/O completion thread ended.
```

```
C:\github\IT-Tage\Workshop\IL\ostress>cmd2

12/04/20 23:06:14.530 [0x000256C8] Total IO waits: 0, Total IO wait time: 0 (ms)
12/04/20 23:06:14.530 [0x000256C8] OSTRESS exiting normally, elapsed time: 00:00:08.574
12/04/20 23:06:14.531 [0x000256C8] RsFx I/O completion thread ended.
```

```
C:\github\IT-Tage\Workshop\IL\ostress>cmd3

12/04/20 23:06:30.607 [0x0001D14C] Total IO waits: 0, Total IO wait time: 0 (ms)
12/04/20 23:06:30.607 [0x0001D14C] OSTRESS exiting normally, elapsed time: 00:00:09.044
12/04/20 23:06:30.608 [0x0001D14C] RsFx I/O completion thread ended.
```



HOW IT WORKS?

- This is the first time that part of the query is run during compilation
  - Therefore, the feature can be called revolutionary

- Improved cardinality estimation enables better operator choice and better estimation of memory requirements

MONITORING

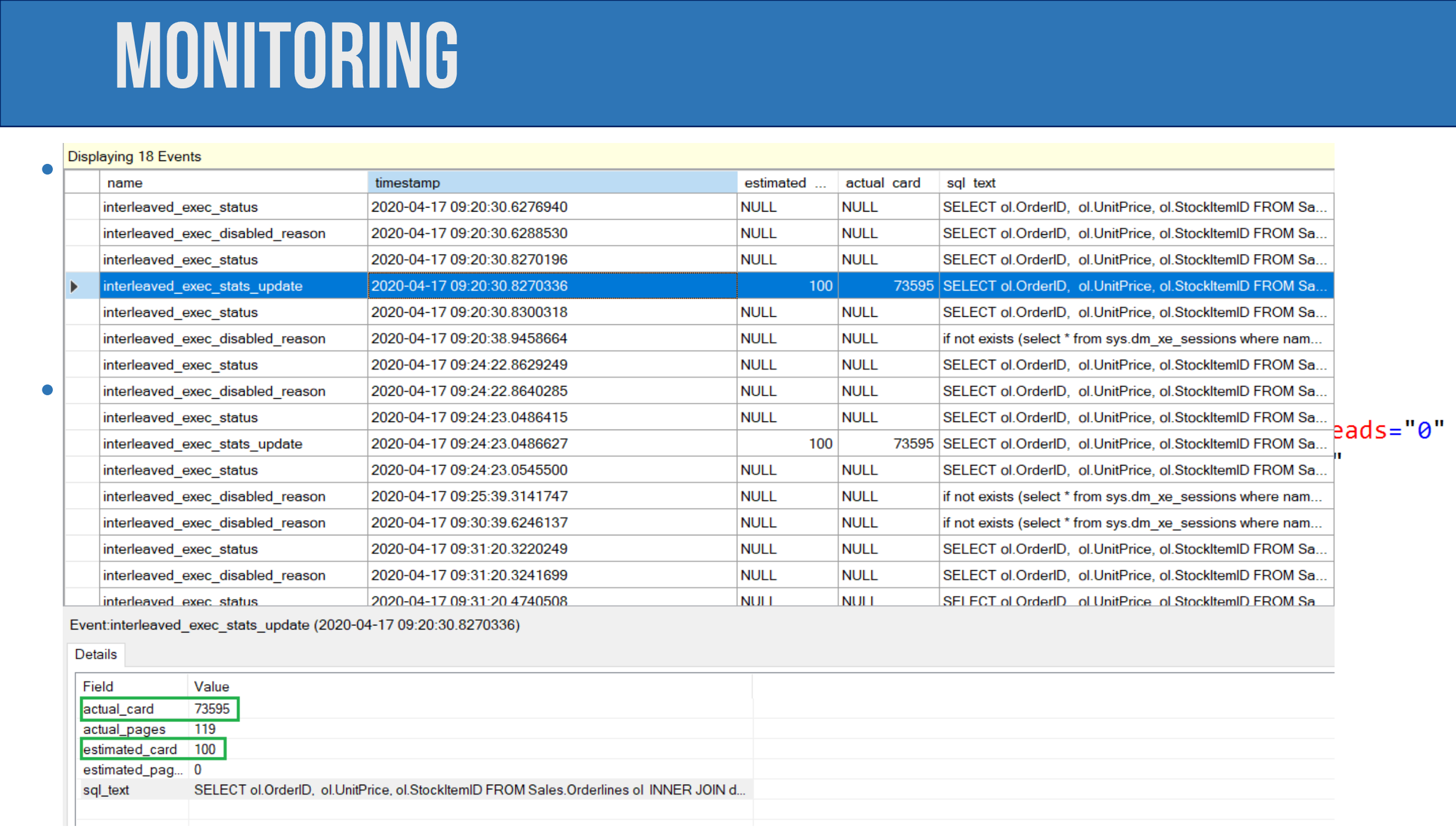
Displaying 18 Events

name	timestamp	estimated ...	actual card	sql text
interleaved_exec_status	2020-04-17 09:20:30.6276940	NULL	NULL	SELECT ol.OrderID, ol.UnitPrice, ol.StockItemID FROM Sa...
interleaved_exec_disabled_reason	2020-04-17 09:20:30.6288530	NULL	NULL	SELECT ol.OrderID, ol.UnitPrice, ol.StockItemID FROM Sa...
interleaved_exec_status	2020-04-17 09:20:30.8270196	NULL	NULL	SELECT ol.OrderID, ol.UnitPrice, ol.StockItemID FROM Sa...
interleaved_exec_stats_update	2020-04-17 09:20:30.8270336	100	73595	SELECT ol.OrderID, ol.UnitPrice, ol.StockItemID FROM Sa...
interleaved_exec_status	2020-04-17 09:20:30.8300318	NULL	NULL	SELECT ol.OrderID, ol.UnitPrice, ol.StockItemID FROM Sa...
interleaved_exec_disabled_reason	2020-04-17 09:20:38.9458664	NULL	NULL	if not exists (select * from sys.dm_xe_sessions where nam...
interleaved_exec_status	2020-04-17 09:24:22.8629249	NULL	NULL	SELECT ol.OrderID, ol.UnitPrice, ol.StockItemID FROM Sa...
interleaved_exec_disabled_reason	2020-04-17 09:24:22.8640285	NULL	NULL	SELECT ol.OrderID, ol.UnitPrice, ol.StockItemID FROM Sa...
interleaved_exec_status	2020-04-17 09:24:23.0486415	NULL	NULL	SELECT ol.OrderID, ol.UnitPrice, ol.StockItemID FROM Sa...
interleaved_exec_stats_update	2020-04-17 09:24:23.0486627	100	73595	SELECT ol.OrderID, ol.UnitPrice, ol.StockItemID FROM Sa...
interleaved_exec_status	2020-04-17 09:24:23.0545500	NULL	NULL	SELECT ol.OrderID, ol.UnitPrice, ol.StockItemID FROM Sa...
interleaved_exec_disabled_reason	2020-04-17 09:25:39.3141747	NULL	NULL	if not exists (select * from sys.dm_xe_sessions where nam...
interleaved_exec_disabled_reason	2020-04-17 09:30:39.6246137	NULL	NULL	if not exists (select * from sys.dm_xe_sessions where nam...
interleaved_exec_status	2020-04-17 09:31:20.3220249	NULL	NULL	SELECT ol.OrderID, ol.UnitPrice, ol.StockItemID FROM Sa...
interleaved_exec_disabled_reason	2020-04-17 09:31:20.3241699	NULL	NULL	SELECT ol.OrderID, ol.UnitPrice, ol.StockItemID FROM Sa...
interleaved_exec_status	2020-04-17 09:31:20.4740508	NULL	NULL	SELECT ol.OrderID, ol.UnitPrice, ol.StockItemID FROM Sa

Event:interleaved_exec_stats_update (2020-04-17 09:20:30.8270336)

Details

Field	Value
actual_card	73595
actual_pages	119
estimated_card	100
estimated_pag...	0
sql_text	SELECT ol.OrderID, ol.UnitPrice, ol.StockItemID FROM Sales.Orderlines ol INNER JOIN d...



LIMITATIONS

- Only for functions for fixed parameters

```
CREATE OR ALTER FUNCTION dbo.Top2OrdersForCustomer(@CustomerId INT)
RETURNS @T TABLE
(ID    INT  NOT NULL)
AS
BEGIN
    INSERT INTO @T SELECT TOP (2) OrderId FROM Sales.Orders WHERE
    CustomerID = @CustomerId
    RETURN
END
GO
SELECT * FROM Sales.Customers c
CROSS APPLY dbo.Top2OrdersForCustomer(c.CustomerID) x
```

LIMITATIONS

- Only SELECT queries can be optimized

```
SELECT ol.OrderID,  ol.UnitPrice, ol.StockItemID
INTO #z
FROM Sales.Orderlines ol
INNER JOIN dbo.SignificantOrders() f1 ON f1.Id = ol.OrderID WHERE PackageTypeID = 7;
```

	name	timestamp	estimated ...	actual card
▶	interleaved_exec_disabled_reason	2020-04-29 13:39:53.3569494	NULL	NULL
	interleaved_exec_status	2020-04-29 13:39:34.8877182	NULL	NULL
	interleaved_exec_stats_update	2020-04-29 13:39:34.8850858	100	73595

Event:interleaved_exec_disabled_reason (2020-04-29 13:39:53.3569494)

Details

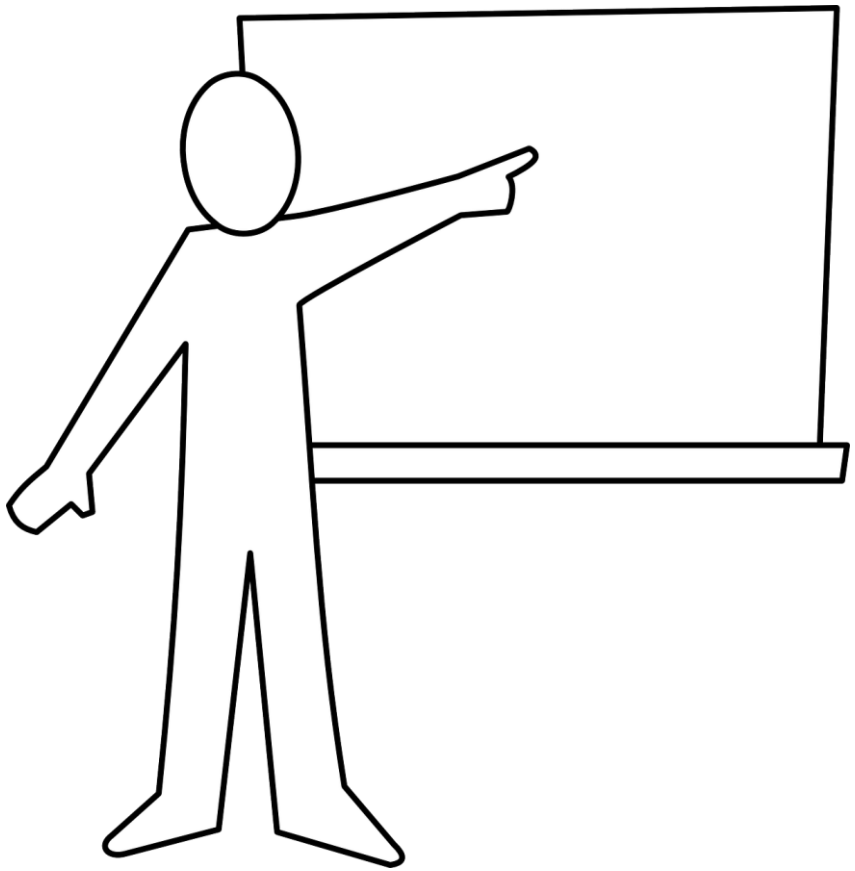
Field	Value
disabled_reason	SelectQueryOnly
sql_text	SELECT ol.OrderID, ol.UnitPrice, ol.StockItemID into #z FROM Sales.Orderlines ol INNER JOIN dbo.SignificantOr...

- Parameter Sniffing

PLAN REGRESSIONS

- It was fine with no nested execution, and now isn't it?
- Regressions are possible for plans that were still acceptable despite a poor MSTVF estimate
  - Usually, other parts of the plan were overestimated, so the MSTVF's underestimation "ironed out the cardinality"
  - The MSTVF cardinality correction broke the workaround

PLAN REGRESSIONS



DEMO

CONFIGURATION

- SQL Server 2017

```
ALTER DATABASE SCOPED CONFIGURATION SET DISABLE_INTERLEAVED_EXECUTION_TVF = OFF;
ALTER DATABASE SCOPED CONFIGURATION SET DISABLE_INTERLEAVED_EXECUTION_TVF = ON;
```

- SQL Server 2019

```
ALTER DATABASE SCOPED CONFIGURATION SET INTERLEAVED_EXECUTION_TVF = ON;
ALTER DATABASE SCOPED CONFIGURATION SET INTERLEAVED_EXECUTION_TVF = OFF;
```

- Auf Abfrage-Ebene

```
OPTION (USE HINT('DISABLE_INTERLEAVED_EXECUTION_TVF'));
```

CONCLUSION

- Useful feature, but with a limited scope
- It only improves cardinality for queries that call MSTVF
- Only works for functions without parameters, or F with fixed parameters
- And only for pure SELECT queries
- The feature can be switched on / off at the database or query level