Free Online Training for Data Professionals.
By the Community, For the Community.

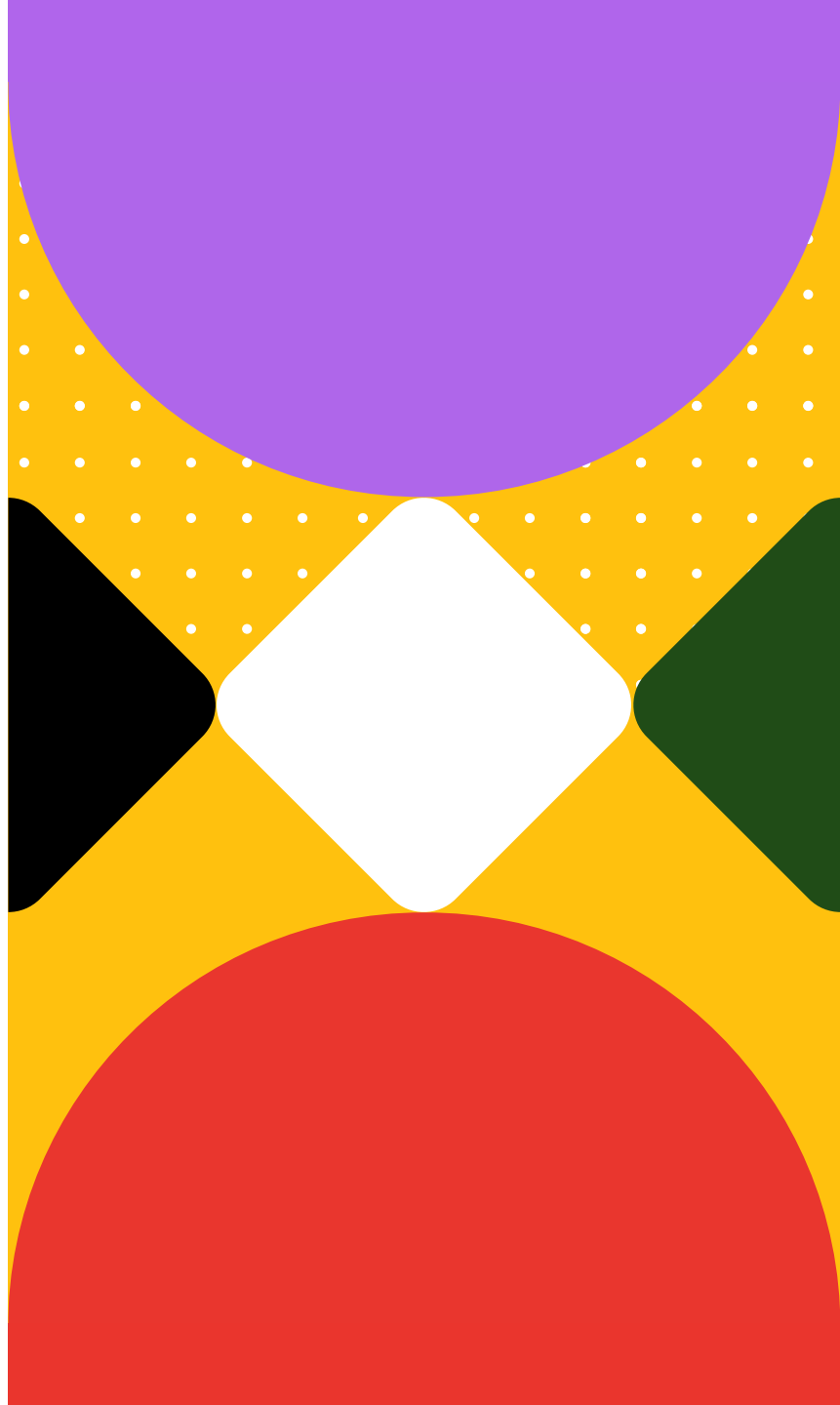
GroupBy Code Of Conduct

The Quick Version

We are dedicated to a harassment-free experience for everyone, regardless of who you are and what makes you *you*. We recognize the right of any individual to attend and participate. Anyone. This is included but not limited to gender identity and expression, sexual orientation, disability, physical appearance, body size, race, religion, or any other classification, affiliation, or label.

- **We do not tolerate harassment in any form.** For the duration of your engagement with GroupBy and its programs, you are expected to act appropriately and to adhere to this Code of Conduct. This includes conduct in-person and online, at the conference itself, as well as any non-conference programs that may include participants: including talks, workshops, parties, on social media, and other online forums. GroupBy participants violating these rules may be sanctioned or expelled without a refund (if that applies) at the discretion of the conference organizers.

- You can review the full policy at: **GroupBy.org/Code-of-Conduct**



GroupBy

Free Online Training for Data Professionals.

By the Community, For the Community.

GroupBy

Media Sponsor:

IDERA.com

Damir Matešić, mag. inf.

- Senior Database Developer @Span.eu

- AD 2018 - Leading Data Events in Croatia
- AD 2019 - Introduced SQL Saturday in Croatia
- AD 2020 - Co-founder & organizer of #Dataweekender…

W: blog.matesic.info

@: damir@matesic.info

in: linkedin.com/in/dmatesic

What’s new?

SQL 2016

- JSON
- System-Versioned (Temporal) Tables
- String functions (STRING_SPLIT, STRING_ESCAPE, FORMATMESSAGE)
- HASHBYTES enhancements
- DATEDIFF_**BIG**
- COMPRESS and DECOMPRESS, AT TIME ZONE
- CREATE OR ALTER & DROP IF EXISTS (a.k.a. DIE)
- Query Store, PolyBase, Stretch Database, Security
- …

SQL 2017

- Adaptive query processing
- String functions (TRIM, TRANSLATE, STRING_AGG, CONCAT_WS…)
- Automatic database tuning
- Resumable online index rebuild
- CLR assemblies
- …

SQL 2019

- String or binary data would be truncated
- UTF8
- Adaptive query processing, part 2
- Accelerated Database Recovery
- Optimizing for sequential key
- Lightweight Statistics
- Memory-Optimized TempDB Metadata
- …

Agenda

- Short overview:
  - Accelerated Database Recovery
  - Optimizing for sequential key
  - Lightweight Statistics
  - Memory-Optimized TempDB Metadata
  - String(s)
  - Intelligent Query Processing*

- *If time allows us

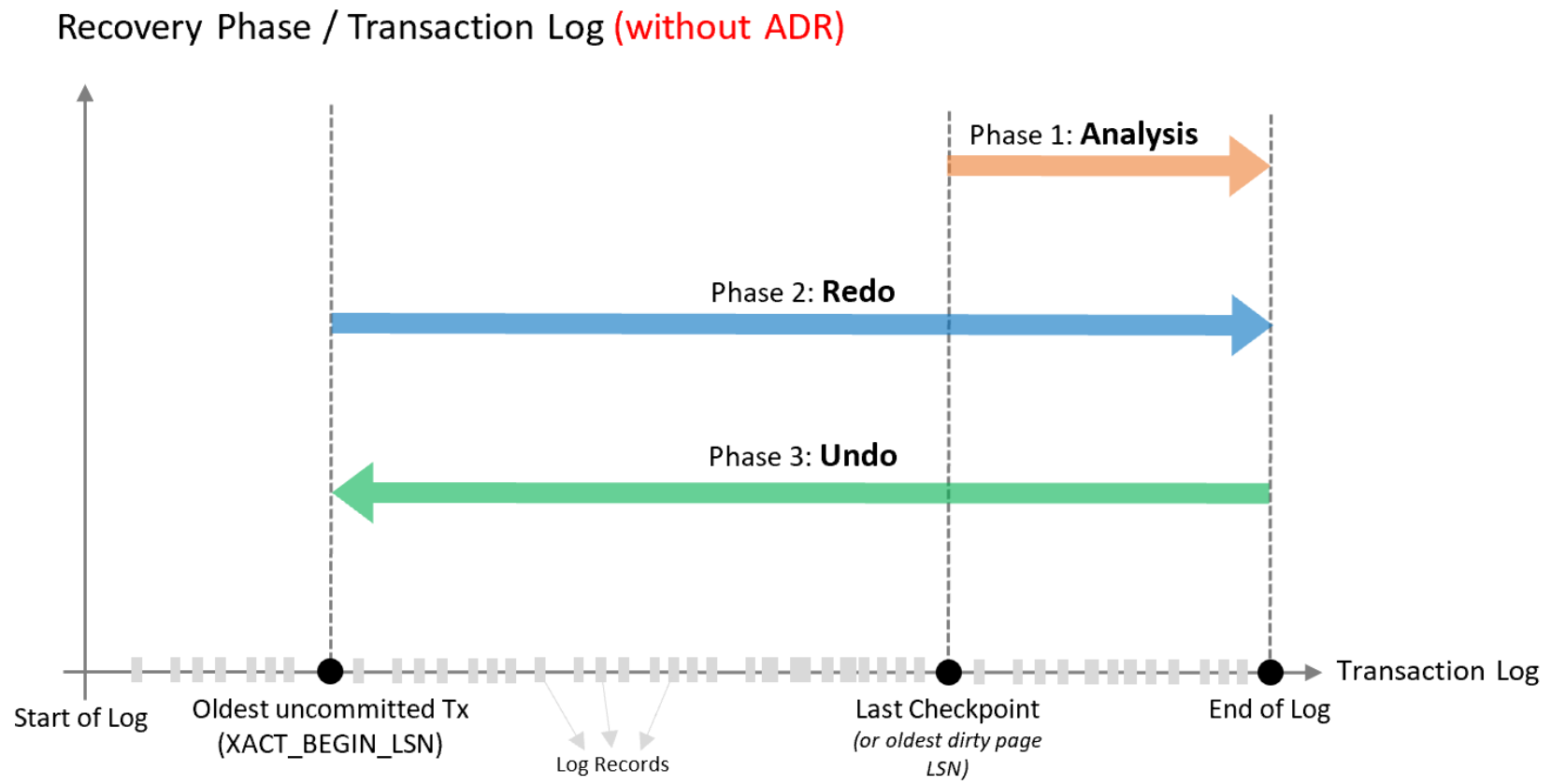
Accelerated Database Recovery (ADR)

ADR

- Current Database Recovery Process
  - Recovery time is proportional to the size of the longest active transaction
  - Transaction log cannot be truncated
- Accelerated Database Recovery
  - Significantly improves database availability especially in the presence of long running transactions
  - Fast and consistent database recovery
  - Instantaneous transaction rollback
  - Aggressive log truncation

ADR

- Current Database Recovery Process



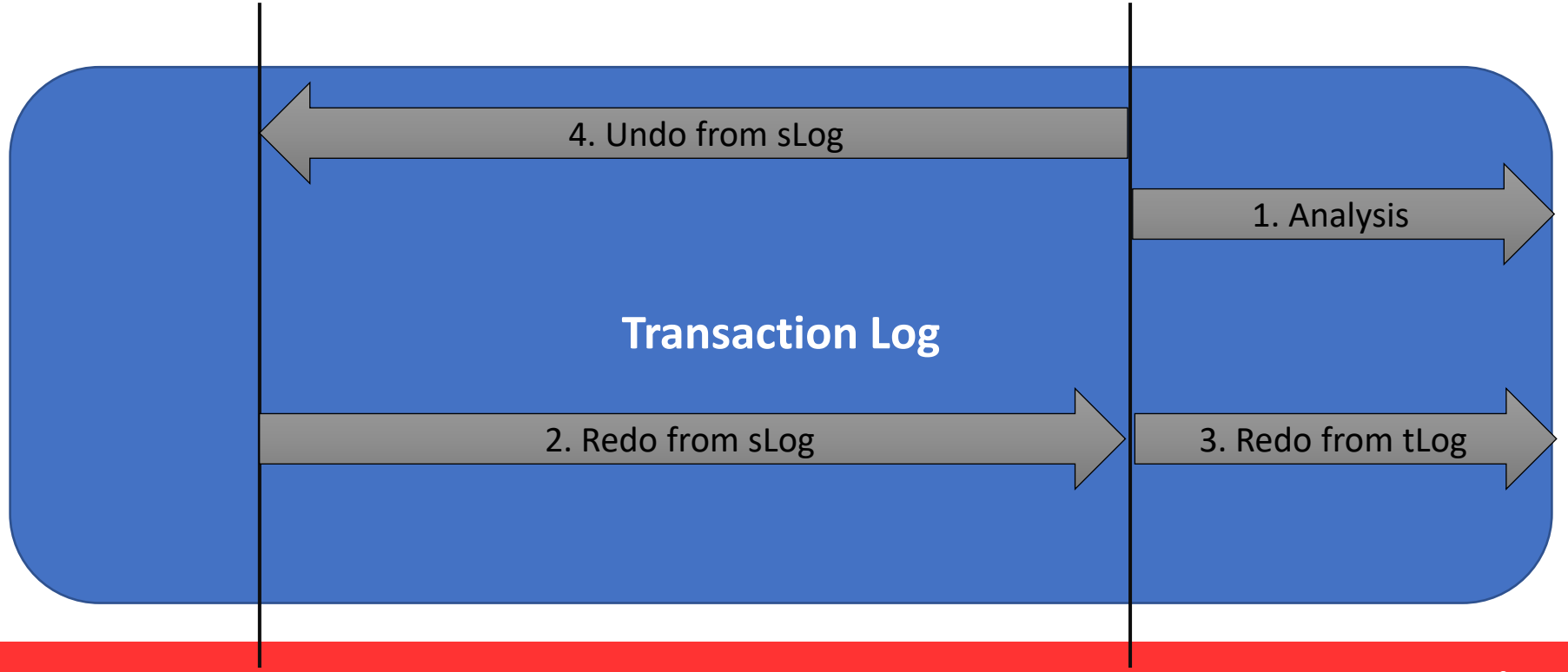
GroupBy

ADR

- Components:
  - **Persisted Version Store (PVS)** - previous versions of data modified by transactions
  - **Logical revert** – active transactions read records from PSV
  - **sLog** – In-memory log stream
  - **Cleaner**

ADR

- Accelerated Database Recovery




DEMO

OPTIMIZE_FOR_SEQUENTIAL_KEY

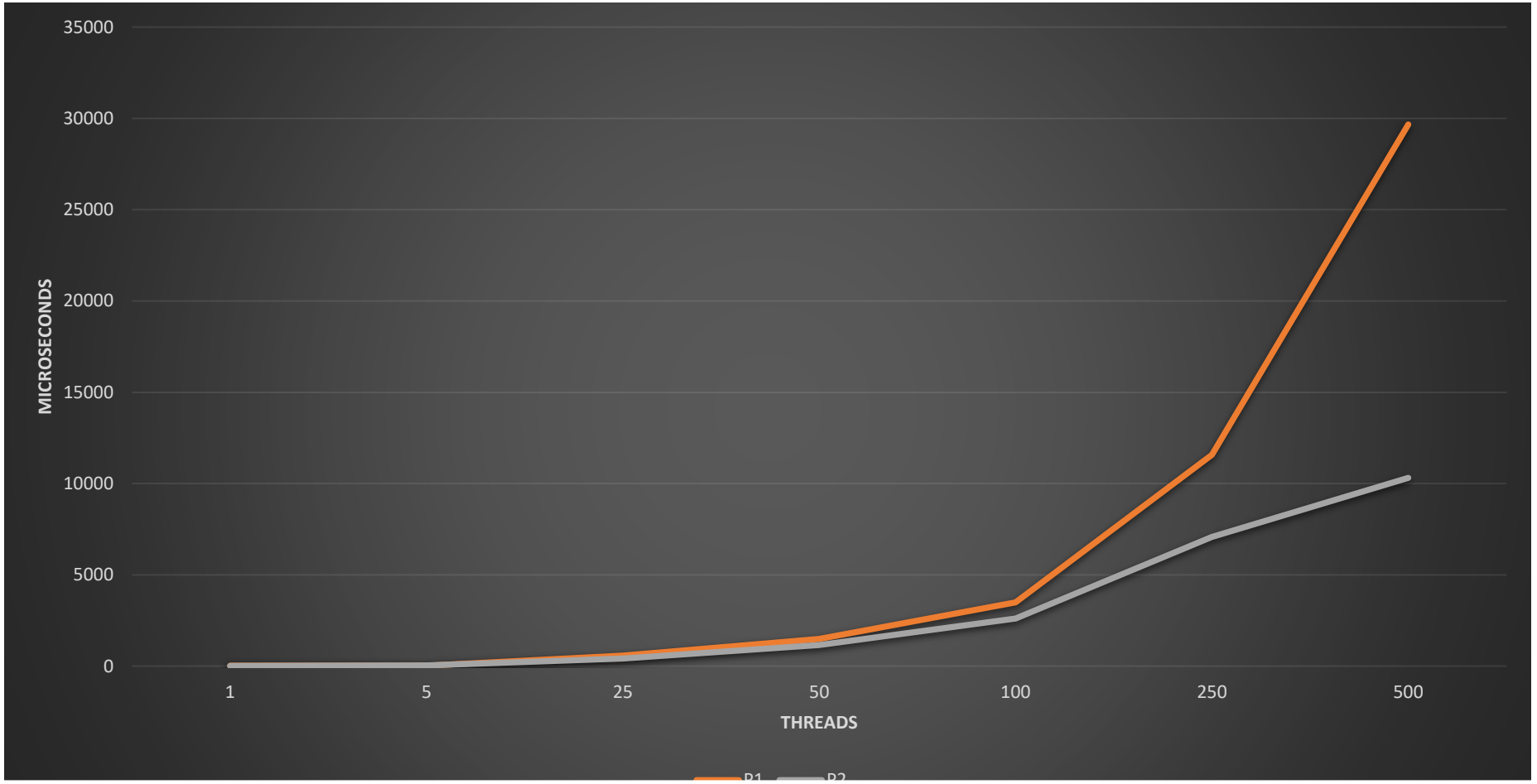
OPTIMIZE_FOR_SEQUENTIAL_KEY

- New option to deal with the last page insert contention
- OPTIMIZE_FOR_SEQUENTIAL_KEY improves throughput for high concurrency inserts into the index (e.g. autoincrement number or uniqueidentifier)

OPTIMIZE_FOR_SEQUENTIAL_KEY

```
CREATE TABLE [dbo].[Table](
  [ID] BIGINT IDENTITY(1,1) NOT NULL,
  [Name] NVARCHAR(128) NOT NULL,
  [SomeNumber] INT NOT NULL,
  CONSTRAINT PK_T1 PRIMARY KEY CLUSTERED
    ([ID]) WITH (OPTIMIZE_FOR_SEQUENTIAL_KEY = ON)
);
```

OPTIMIZE_FOR_SEQUENTIAL_KEY



AVG(Query plan duration), 50x „Insert 50 rows”; parallel thread from 1 to 500


DEMO

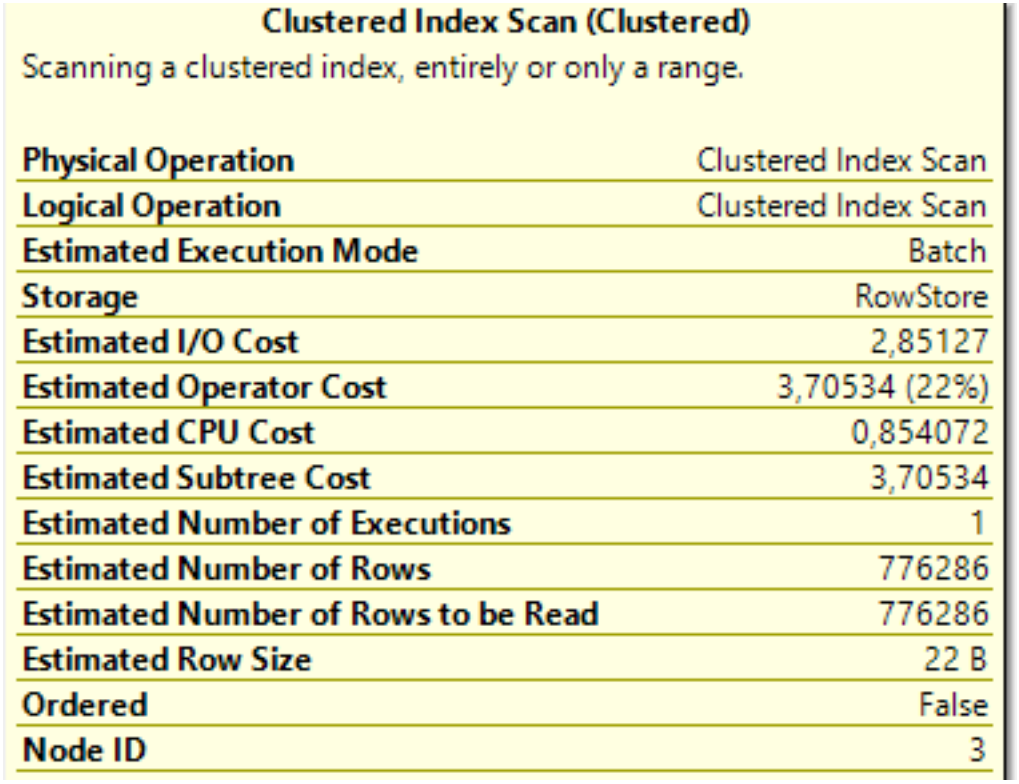
Lightweight Statistics

Lightweight Statistics sys.dm_exec_query_plan_stats

- All plans are estimated!
- How to get metrics?
- sys.dm_exec_query_plan_stats -> actual plan, the last one run on the system

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

Lightweight Statistics
sys.dm_exec_query_plan_stats

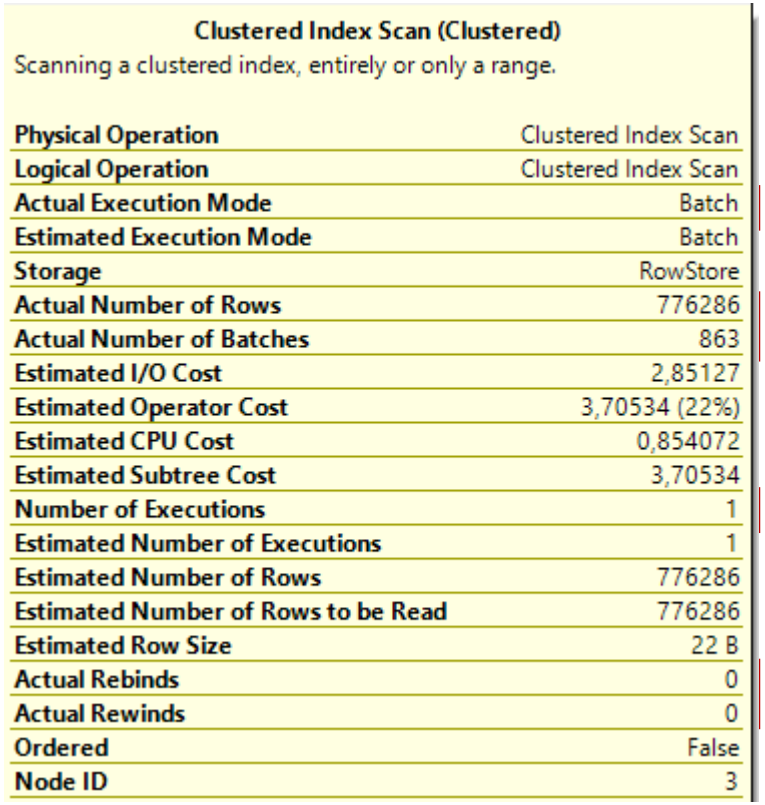


Clustered Index Scan (Clustered)

Scanning a clustered index, entirely or only a range.

Physical Operation	Clustered Index Scan
Logical Operation	Clustered Index Scan
Estimated Execution Mode	Batch
Storage	RowStore
Estimated I/O Cost	2,85127
Estimated Operator Cost	3,70534 (22%)
Estimated CPU Cost	0,854072
Estimated Subtree Cost	3,70534
Estimated Number of Executions	1
Estimated Number of Rows	776286
Estimated Number of Rows to be Read	776286
Estimated Row Size	22 B
Ordered	False
Node ID	3

ESTIMATED PLAN



Clustered Index Scan (Clustered)

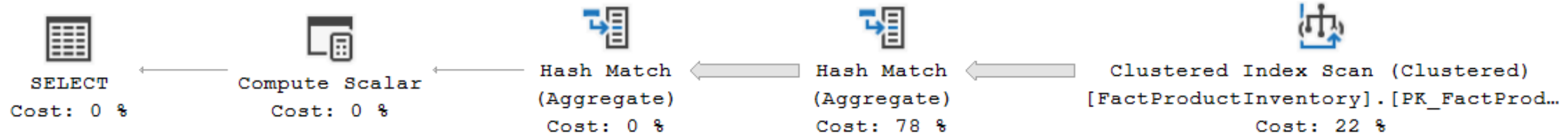
Scanning a clustered index, entirely or only a range.

Physical Operation	Clustered Index Scan
Logical Operation	Clustered Index Scan
Actual Execution Mode	Batch
Estimated Execution Mode	Batch
Storage	RowStore
Actual Number of Rows	776286
Actual Number of Batches	863
Estimated I/O Cost	2,85127
Estimated Operator Cost	3,70534 (22%)
Estimated CPU Cost	0,854072
Estimated Subtree Cost	3,70534
Number of Executions	1
Estimated Number of Executions	1
Estimated Number of Rows	776286
Estimated Number of Rows to be Read	776286
Estimated Row Size	22 B
Actual Rebinds	0
Actual Rewinds	0
Ordered	False
Node ID	3

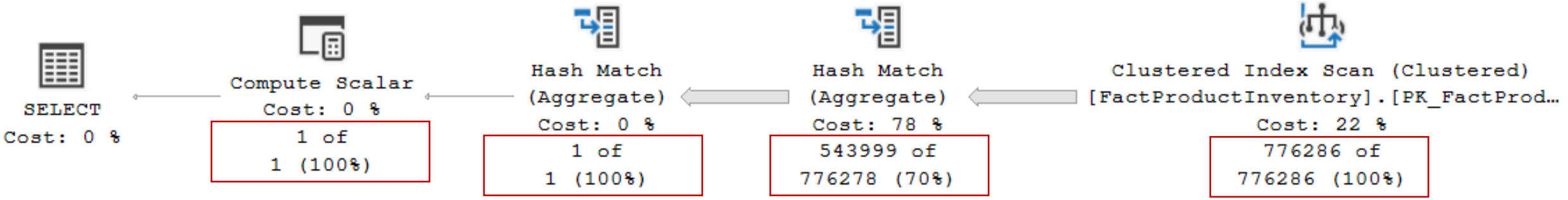
„ACTUAL” PLAN

Lightweight Statistics
sys.dm_exec_query_plan_stats

ESTIMATED PLAN



„ACTUAL" PLAN



GroupBy


DEMO

Memory-Optimized TempDB Metadata

Memory-Optimized TempDB Metadata

- Server feature -> SERVER RESTART
- Key tempdb system tables become SCHEMA_ONLY memory optimized tables
- Latch and lock free
- Just metadata NOT user data!

```
ALTER SERVER CONFIGURATION SET MEMORY_OPTIMIZED TEMPDB_METADATA = ON;
ALTER SERVER CONFIGURATION SET MEMORY_OPTIMIZED TEMPDB_METADATA = OFF;
```

Memory-Optimized TempDB Metadata

- What can you break?
  - columnstore indexes
  - sp_estimate_data_compression_savings
  - transactions with memory-optimized tables

Memory-Optimized TempDB Metadata

ostress.exe -E -dTestDb -Q"EXEC dbo.P" -MSSQLSERVER -r100 -n100 -q

ALTER SERVER CONFIGURATION SET MEMORY_OPTIMIZED TEMPDB_METADATA = OFF;

```
12/09/19 13:52:15.603 [0x00004CA0] Starting query execution...
12/09/19 13:52:15.606 [0x00004CA0]  BETA: Custom CLR Expression support enabled.
12/09/19 13:52:15.606 [0x00004CA0] Creating 100 thread(s) to process queries
12/09/19 13:52:15.614 [0x00004CA0] Worker threads created, beginning execution...
12/09/19 13:54:03.602 [0x00004CA0] Total IO waits: 0, Total IO wait time: 0 (ms)
12/09/19 13:54:03.602 [0x00004CA0] OSTRESS exiting normally, elapsed time: 00:01:48.216
```

ALTER SERVER CONFIGURATION SET MEMORY_OPTIMIZED TEMPDB_METADATA = ON;

```
12/09/19 13:50:44.286 [0x000026BC] Starting query execution...
12/09/19 13:50:44.289 [0x000026BC]  BETA: Custom CLR Expression support enabled.
12/09/19 13:50:44.289 [0x000026BC] Creating 100 thread(s) to process queries
12/09/19 13:50:44.297 [0x000026BC] Worker threads created, beginning execution...
12/09/19 13:51:29.756 [0x000026BC] Total IO waits: 0, Total IO wait time: 0 (ms)
12/09/19 13:51:29.756 [0x000026BC] OSTRESS exiting normally, elapsed time: 00:00:45.688
```

Memory-Optimized TempDB Metadata

ostress.exe -E -dTestDb -Q"EXEC dbo.P" -MSSQLSERVER -r100 -n100 -q

ALTER SERVER CONFIGURATION SET MEMORY_OPTIMIZED TEMPDB_METADATA = OFF;

Results | Messages

	wait_time	wait_type	total_elapsed_time	cpu_time	text	client_net_address	num_reads	num_writes
1	11	PAGELATCH_EX	1003	22	CREATE PROCEDURE dbo.P AS DECLARE @cnt INT =...	<local machine>	17	17
2	11	PAGELATCH_EX	421	8	CREATE PROCEDURE dbo.P AS DECLARE @cnt INT =...	<local machine>	16	15
3	10	PAGELATCH_EX	676	18	CREATE PROCEDURE dbo.P AS DECLARE @cnt INT =...	<local machine>	16	15
4	10	PAGELATCH_EX	654	18	CREATE PROCEDURE dbo.P AS DECLARE @cnt INT =...	<local machine>	16	15
5	10	PAGELATCH_EX	264	6	CREATE PROCEDURE dbo.P AS DECLARE @cnt INT =...	<local machine>	16	15
6	9	PAGELATCH_EX	299	8	CREATE PROCEDURE dbo.P AS DECLARE @cnt INT =...	<local machine>	16	15
7	9	PAGELATCH_EX	806	21	CREATE PROCEDURE dbo.P AS DECLARE @cnt INT =...	<local machine>	16	15
8	9	PAGELATCH_EX	126	2	CREATE PROCEDURE dbo.P AS DECLARE @cnt INT =...	<local machine>	17	16

ALTER SERVER CONFIGURATION SET MEMORY_OPTIMIZED TEMPDB_METADATA = ON;

Results | Messages

	wait_time	wait_type	total_elapsed_time	cpu_time	text	client_net_address	num_reads	num_writes
1	0	MEMORY_ALLOCATION_EXT	239	19	CREATE PROCEDURE dbo.P AS DECLARE @cnt INT =...	<local machine>	43	42


DEMO

String(s)

Strings – SQL 2016

- STRING_SPLIT
  - table-valued function
  - splitting string values by a separator
- STRING_ESCAPE
  - escapes special characters
  - by any rule if this rule is JSON
- FORMATMESSAGE (2008)
  - FORMATMESSAGE ({ msg_number | '**msg_string**' } , [param_value [,...n]])

Strings – SQL 2017

- TRIM -> LTRIM(RTRIM(String)) = TRIM(String)
- STRING_AGG
  - string aggregation using a separator
- TRANSLATE
  - replaces multiple characters inside the given string value
- CONCAT_WS

Strings – SQL 2019

String or binary data would be truncated

- Old error message
- Msg 8152, Level 16, State 30, Line 18
- String or binary data would be truncated.
- New error message
- Msg 2628, Level 16, State 1, Line 35
- String or binary data would be truncated in table ‘SomeDatabase.dbo.SomeTable ', column ‘Col'. Truncated value: ‘XXX’

Strings – SQL 2019

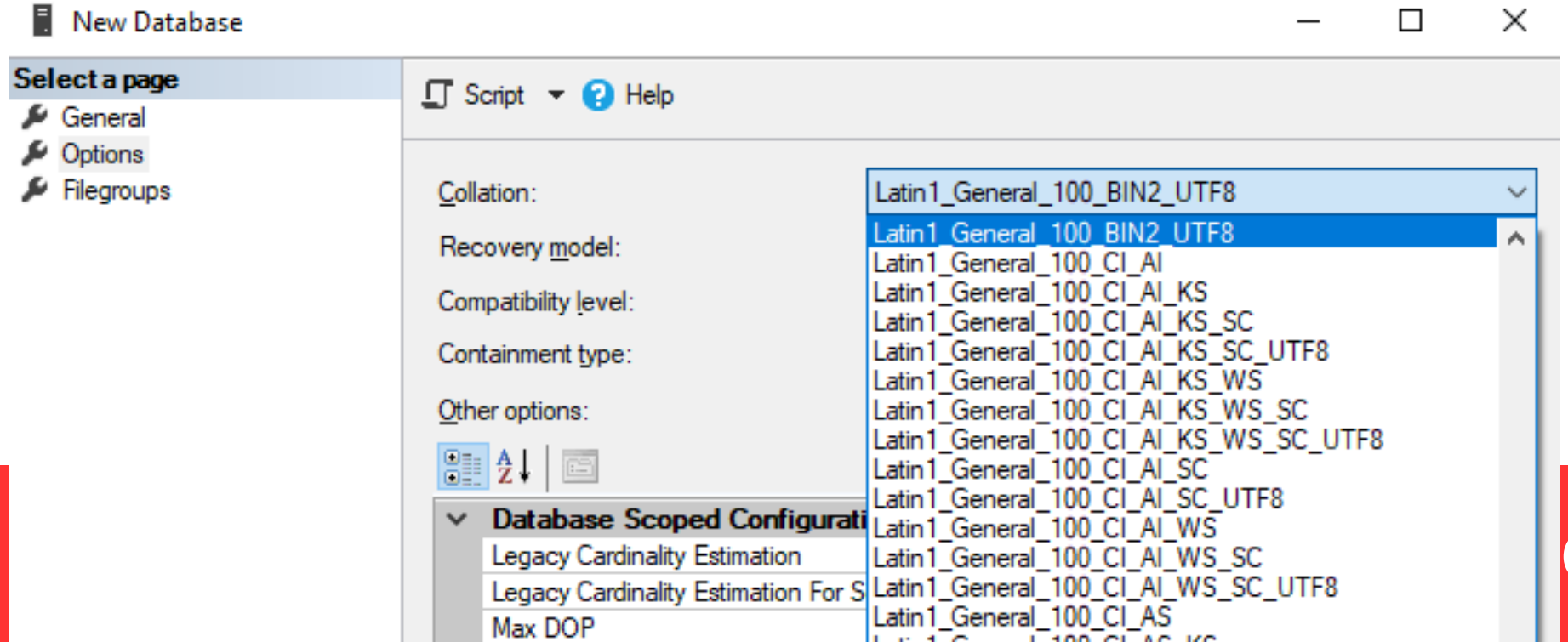
String or binary data would be truncated

- Msg 8152, Level 16, State 30, Line 18
- Msg 2628, Level 16, State 1, Line 35

```
ALTER DATABASE SCOPED CONFIGURATION SET VERBOSE_TRUNCATION_WARNINGS = OFF;
```

Strings – SQL 2019
UTF8

- Allow application(s) internationalization without converting all strings to Unicode
- Implemented as new collation -> 1.553 new collations)



GroupBy

Strings – SQL 2019
UTF8

- SQL 2019 supports UTF 8 collation -> SQL 2019 підтримує сортування UTF 8

```
USE NONUTF8
GO

DECLARE @v VARCHAR(100) = 'SQL 2019 підтримує сортування UTF 8';
SELECT @v AS String, DATALENGTH(@v) AS DataLengthValue;
DECLARE @nv NVARCHAR(100) = N'SQL 2019 підтримує сортування UTF 8';
SELECT @nv AS String, DATALENGTH(@nv) AS DataLengthValue;
GO
```

150 %

Results | Messages

	String	DataLengthValue
1	SQL 2019 ????????? ?????????? UTF 8	35

	String	DataLengthValue
1	SQL 2019 підтримує сортування UTF 8	70

```
USE UTF8;
GO

DECLARE @8v VARCHAR(100) = 'SQL 2019 підтримує сортування UTF 8';
SELECT @8v AS String, DATALENGTH(@8v) AS DataLengthValue;
DECLARE @8nv NVARCHAR(100) = N'SQL 2019 підтримує сортування UTF 8';
SELECT @8nv AS String, DATALENGTH(@8nv) AS DataLengthValue;
GO
```

50 %

Results | Messages

	String	DataLengthValue
1	SQL 2019 підтримує сортування UTF 8	54

	String	DataLengthValue
1	SQL 2019 підтримує сортування UTF 8	70


DEMO

Intelligent Query Processing

Intelligent Query Processing

- T-SQL Scalar UDF Inlining
- Approximate Query Processing
- Batch and Row Mode Memory Grant Feedback
- Batch Mode on Rowstore
- Table Variable Deferred Compilation
- Adaptive joins
- Interleaved execution

Thank you