Polish SQL Server User Group

Data Community

SQL Day 2017

GOLD SPONSORS

Atos

Profisee®

Zerto

SILVER SPONSORS

PROMISE
GRUPA APN PROMISE

ELITMIND

ITMAGINATION
Imagine it. Done.

THEOBALD SOFTWARE

BRONZE SPONSORS

it KONTRAKT®

JCommerce

IT
Krajowy Rejestr Długów

KSIĄŻKI IT Z CAŁEGO ŚWIATA
NOVATECH
www.novatech.com.pl

PEPCO

dbWatch
DATABASE CONTROL

querona

Data Scientist as a Service

UBS

VOLVO
Volvo Group IT

STRATEGIC PARTNER

Microsoft

Enhancements that will make your SQL database engine roar

SQL Server Tiger Team

Microsoft

Pedro Lopes

@sqlpto
pedro.lopes@microsoft.com

Senior Program Manager

Focused on SQL Server Relational Engine

7+ years at Microsoft

14+ years with SQL Server

Using right SQL Features for the workload

- Business grows from small scale to mission critical
  - Initial deployment targets Standard Edition to keep the cost low
  - Application uses features available in Standard Edition
  - As business grows and migrates to EE, customers reluctant to modify application
  - Example – using 'disk-based' tables for OLTP workloads
- Packaged Applications from ISVs
  - Target Enterprise and lower Editions to reach customers across spectrum
  - Preferred Development Model – Single Code Base
  - Common Application Development Strategy –
    - Use only the features available in ALL supported editions
    - This limitation leads to non-optimal use of features for the targeted workload.
    - Example – not using In-Memory OLTP for transactional workload

Solution - Right SQL Features for the workload

- Common Programmability Surface Area (CPSA) across SQL Editions
  - Optimal Design: Enables application to use the right features for the workload
  - Freedom to deploy the same application to any SQL Server Edition
  - Choice of Edition – Based on Performance, scalability and Availability requirements

Feature	RTM				SP1			
	Standard	Web	Express	Local DB	Standard	Web	Express	Local DB
Row-level security	Yes	No	No	No	Yes	Yes	Yes	Yes
Dynamic Data Masking	Yes	No	No	No	Yes	Yes	Yes	Yes
Change data capture*	No	No	No	No	Yes	Yes	No*	No*
Database snapshot	No	No	No	No	Yes	Yes	Yes	Yes
Columnstore	No	No	No	No	Yes	Yes	Yes	Yes
Partitioning	No	No	No	No	Yes	Yes	Yes	Yes
Compression	No	No	No	No	Yes	Yes	Yes	Yes
In Memory OLTP	No	No	No	No	Yes	Yes	Yes	No**
Always Encrypted	No	No	No	No	Yes	Yes	Yes	Yes
PolyBase	No	No	No	No	Yes	Yes	Yes	No
Fine grained auditing	No	No	No	No	Yes	Yes	Yes	Yes
Multiple filestream containers	No	No	No	No	Yes	Yes	Yes	Yes

* Requires SQL Server Agent which is not part of SQL Server Express Editions.

** Requires creating filestream file groups which is not possible in Local DB due to insufficient permissions.

SQL Server Tiger Team

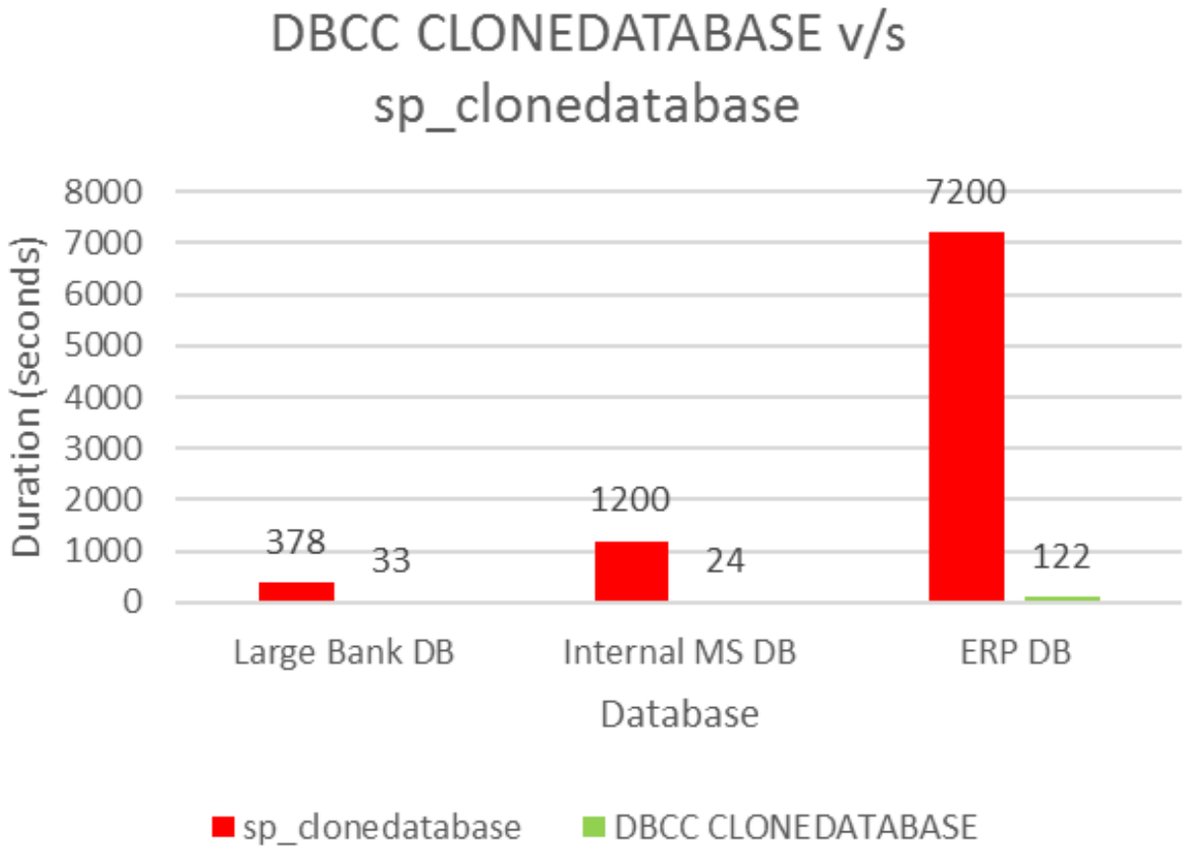
Storage Engine

DBCC CLONEDATABASE

- Fast, minimally invasive and consistent.
- Introduced first in SQL Server 2014 SP2.
- Introduced & Enhanced in SQL Server 2016 SP1 with support for:
  - CLR Objects
  - Filestream/FileTable Objects
  - In-Memory tables
  - Query Store (Persisted Query Plan Store)

DBCC CLONEDATABASE

Database	Number of Objects	SSMS	sp_clonedatabase	DBCC CLONEDATABASE
Large Bank DB	20453	Scripting error	378 seconds*	33 seconds (11x)
Internal MS DB	80819	OOM	1200 seconds*	24 seconds (50x)
ERP DB	1008002	OOM	7200 seconds*	122 seconds (60x)



SQL Server Tiger Team

DBCC CLONEDATABASE

DEMO

DBCC CLONEDATABASE

- Default CLONE includes schema, statistics and query store data.

```
-- Default settings generate cloned database with statistics and query store metadata.
DBCC CLONEDATABASE (source_database_name, target_database_name)

-- Schema and query store only clone
DBCC CLONEDATABASE (source_database_name, target_database_name) WITH NO_STATISTICS
```

Recommended for functional tests – no statistics means no data shown in histograms

```
-- Schema and statistics only clone
DBCC CLONEDATABASE (source_database_name, target_database_name) WITH NO_QUERYSTORE
```

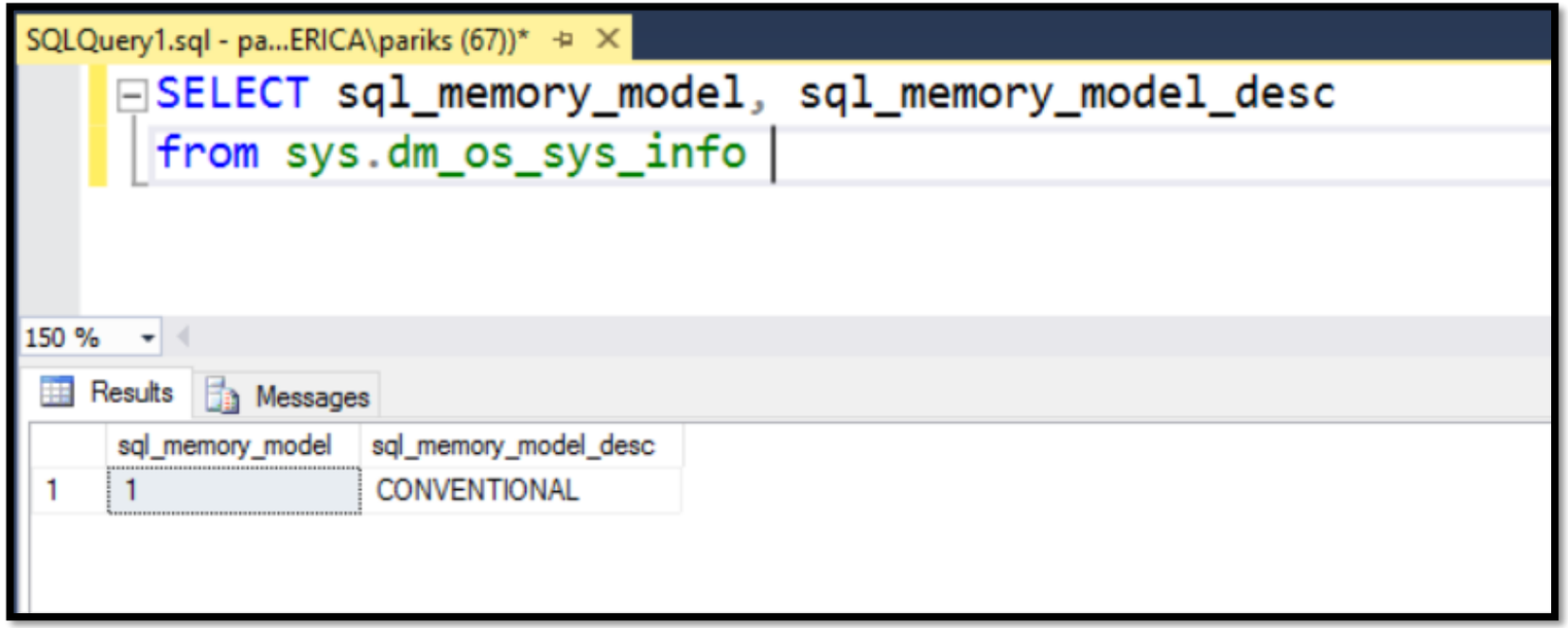
Recommended for perf testing with existing set of heavy hitter queries

```
-- Schema only clone
DBCC CLONEDATABASE (source_database_name, target_database_name) WITH
NO_STATISTICS, NO_QUERYSTORE
```

Recommended for dev/test environment – no query store used in SQL Server 2014 SP2

Supportability Improvements

- New *sql_memory_model*, *sql_memory_model_desc* columns in DMV **sys.dm_os_sys_info** to allow DBAs to programmatically identify **Lock Pages in Memory** (LPIM) privilege to SQL Service account.




```
1 = Conventional Memory Model
2 = Lock Pages in Memory
3 = Large Pages in Memory
```

Supportability Improvements

- New column *instant_file_initialization_enabled* in DMV **sys.dm_server_services** to allow DBAs to programmatically identify **Instant File initialization** (IFI) to SQL Service account.

SQLQuery1.sql - pa...ERICA\pariks (67))*

```
SELECT servicename,instant_file_initialization_enabled
from sys.dm_server_services
```

150 %

Results | Messages

	servicename	instant_file_initialization_enabled
1	SQL Server (MSSQLSERVER)	N
2	SQL Server Agent (MSSQLSERVER)	NULL
3	SQL Full-text Filter Daemon Launcher (MSSQLSERVER)	NULL

Supportability Improvements

- Removing noisy In-Memory logging messages from Errorlog.

2016 (RTM) - 13.0.1601.5	2016-11-03 14:32:55.20 spid4s [INFO] HkHostDbCtxt::Initialize(): Database ID: [1] 'master'. XTP Engine version is 0.0. 2016-11-03 14:32:55.22 spid4s [INFO] HkHostDbCtxt::Initialize(): Database ID: [1] 'master'. XTP Engine version is 0.0. 2016-11-03 14:32:55.39 spid4s [INFO] HkHostDbCtxt::Initialize(): Database ID: [1] 'master'. XTP Engine version is 0.0. 2016-11-03 14:32:57.45 spid14s [INFO] HkHostDbCtxt::Initialize(): Database ID: [4] 'msdb'. XTP Engine version is 0.0. 2016-11-03 14:32:57.45 spid16s [INFO] HkHostDbCtxt::Initialize(): Database ID: [5] 'WideWorldImporters'. XTP Engine version is 0.0. 2016-11-03 14:32:57.45 spid6s [INFO] HkHostDbCtxt::Initialize(): Database ID: [32767] 'mssqlsystemresource'. XTP Engine version is 0.0. 2016-11-03 14:32:57.45 spid17s [INFO] HkHostDbCtxt::Initialize(): Database ID: [6] 'WideWorldImportersDW'. XTP Engine version is 0.0. 2016-11-03 14:32:57.49 spid16s [INFO] HkHostDbCtxt::Initialize(): Database ID: [5] 'WideWorldImporters'. XTP Engine version is 2.9. 2016-11-03 14:32:57.49 spid17s [INFO] HkHostDbCtxt::Initialize(): Database ID: [6] 'WideWorldImportersDW'. XTP Engine version is 2.9. 2016-11-03 14:32:57.50 spid14s [INFO] HkHostDbCtxt::Initialize(): Database ID: [4] 'msdb'. XTP Engine version is 0.0. 2016-11-03 14:32:57.56 spid6s [INFO] HkHostDbCtxt::Initialize(): Database ID: [32767] 'mssqlsystemresource'. XTP Engine version is 0.0. 2016-11-03 14:32:57.67 spid14s [INFO] HkHostDbCtxt::Initialize(): Database ID: [4] 'msdb'. XTP Engine version is 0.0. 2016-11-03 14:32:57.86 spid6s [INFO] HkHostDbCtxt::Initialize(): Database ID: [32767] 'mssqlsystemresource'. XTP Engine version is 0.0. 2016-11-03 14:32:58.02 spid17s [INFO] HkHostDbCtxt::Initialize(): Database ID: [6] 'WideWorldImportersDW'. XTP Engine version is 2.9. 2016-11-03 14:32:58.05 spid16s [INFO] HkHostDbCtxt::Initialize(): Database ID: [5] 'WideWorldImporters'. XTP Engine version is 2.9. 2016-11-03 14:32:58.49 spid19s [INFO] Database ID: [5]. XTP page allocations bound to resource pool 'default'. 2016-11-03 14:32:58.49 spid18s [INFO] Database ID: [6]. XTP page allocations bound to resource pool 'default'. 2016-11-03 14:32:58.49 spid19s [INFO] HostCommonStorage::AddContainer(): Database ID: [5]. Adding container 'E:\MSSQL13.RTMENT\WideWorldImporters_InMemory_Data_1\$HKv2\' to hekaton storage. 2016-11-03 14:32:58.49 spid18s [INFO] HostCommonStorage::AddContainer(): Database ID: [6]. Adding container 'E:\MSSQL13.RTMENT\WideWorldImportersDW_InMemory_Data_1\$HKv2\' to hekaton storage. 2016-11-03 14:32:58.60 spid19s [INFO] RootFileDeserialize(): Database ID: [5]. Root file: {3E231B6B-9EF1-4AE3-A1DB-BD01DC866DD9}, version: 2.9, watermark: 40, RecoveryLsn: 00000272:00005EC4:0002, RecoveryCheckpointId: 4, RecoveryCheckpointTimestamp: 0xb 2016-11-03 14:32:58.60 spid19s [INFO] Database ID: [5]. Deleting unrecoverable checkpoint table row (id: 2). 2016-11-03 14:32:58.66 spid18s [INFO] RootFileDeserialize(): Database ID: [6]. Root file: {01177532-CE1C-4F26-BBFD-08AC9719C58E}, version: 2.9, watermark: 160, RecoveryLsn: 00000040:0000017E:0002, RecoveryCheckpointId: 5, RecoveryCheckpointTimestamp: 0x11 2016-11-03 14:32:58.66 spid18s [INFO] Database ID: [6]. Deleting unrecoverable checkpoint table row (id: 3). 2016-11-03 14:32:58.79 spid6s [INFO] HkHostDbCtxt::Initialize(): Database ID: [3] 'model'. XTP Engine version is 0.0. 2016-11-03 14:32:58.89 spid6s [INFO] HkHostDbCtxt::Initialize(): Database ID: [3] 'model'. XTP Engine version is 0.0. 2016-11-03 14:32:59.34 spid6s [INFO] HkHostDbCtxt::Initialize(): Database ID: [3] 'model'. XTP Engine version is 0.0. 2016-11-03 14:32:59.95 spid19s [INFO] HkCkptLoadInternalEx(): Database ID: [5]. Root file: {3E231B6B-9EF1-4AE3-A1DB-BD01DC866DD9}, watermark: 40, RecoveryLsn: 00000272:00005EC4:0002, RecoveryCheckpointId: 4, RecoveryCheckpointTimestamp: 0xb 2016-11-03 14:33:00.11 spid18s [INFO] HkCkptLoadInternalEx(): Database ID: [6]. Root file: {01177532-CE1C-4F26-BBFD-08AC9719C58E}, watermark: 160, RecoveryLsn: 00000040:0000017E:0002, RecoveryCheckpointId: 5, RecoveryCheckpointTimestamp: 0x11 2016-11-03 14:33:03.51 spid6s [INFO] HkHostDbCtxt::Initialize(): Database ID: [2] 'tempdb'. XTP Engine version is 0.0. 2016-11-03 14:33:03.60 spid6s [INFO] HkHostDbCtxt::Initialize(): Database ID: [2] 'tempdb'. XTP Engine version is 0.0. 2016-11-03 14:33:03.79 spid6s [INFO] HkHostDbCtxt::Initialize(): Database ID: [2] 'tempdb'. XTP Engine version is 0.0. 2016-11-03 14:33:07.68 spid19s [INFO] HkRecoverFromLog(): Database ID: [5]. Log recovery scan from 00000272:00005EC4:0002 to 00000272:000062B0:0001. 2016-11-03 14:33:07.68 spid19s [INFO] HkRecoverFromLog(): Database ID: [5]. Log recovery open segment scan from 00000272:00005EC4:0002 to 00000272:000062B0:0001. 2016-11-03 14:33:07.68 spid19s [INFO] HkRecoverFromLog(): Database ID: [5]. Log recovery open segment scan completed at 00000272:00006260:0001. 2016-11-03 14:33:07.68 spid19s [INFO] HkDatabaseUpdateCheckpointingState(): Database ID: [5]. Future checkpoints are enabled now (RecoveryLSN: 00000272:00005EC4:0002 UndeployLSN: 00000000:00000000:0000). 2016-11-03 14:33:07.95 spid16s [INFO] HkHostStartController(): Database ID: [5]. IsAcceptLogMode=false IsTransitionToLogging=false. 2016-11-03 14:33:07.96 spid16s [INFO] HkHostControllerCtxtImpl::StartController(): Database ID: [5]. Starting controller thread on a dedicated scheduler. 2016-11-03 14:33:07.96 spid19s [INFO] HkHostCheckpointController::DoWork(): Database ID: [5]. Controller scan started at 00000272:00005EC4:0001 2016-11-03 14:33:16.92 spid18s [INFO] HkRecoverFromLog(): Database ID: [6]. Log recovery scan from 00000040:0000017E:0002 to 00000040:000004C0:0001. 2016-11-03 14:33:16.92 spid18s [INFO] HkRecoverFromLog(): Database ID: [6]. Log recovery open segment scan from 00000040:0000017E:0002 to 00000040:000004C0:0001. 2016-11-03 14:33:16.92 spid18s [INFO] HkRecoverFromLog(): Database ID: [6]. Log recovery open segment scan completed at 00000040:00000468:0001. 2016-11-03 14:33:16.92 spid18s [INFO] HkDatabaseUpdateCheckpointingState(): Database ID: [6]. Future checkpoints are enabled now (RecoveryLSN: 00000040:0000017E:0002 UndeployLSN: 00000000:00000000:0000). 2016-11-03 14:33:17.14 spid17s [INFO] HkHostStartController(): Database ID: [6]. IsAcceptLogMode=false IsTransitionToLogging=false. 2016-11-03 14:33:17.14 spid17s [INFO] HkHostControllerCtxtImpl::StartController(): Database ID: [6]. Starting controller thread on a dedicated scheduler. 2016-11-03 14:33:17.14 spid14s [INFO] HkHostCheckpointController::DoWork(): Database ID: [6]. Controller scan started at 00000040:0000017E:0001
2016 (SP1) - 13.0.4001.0	2016-11-03 14:56:29.00 spid16s [INFO] HkHostDbCtxt::Initialize(): Database ID: [5] 'WideWorldImporters'. XTP Engine version is 2.9. 2016-11-03 14:56:29.03 spid17s [INFO] HkHostDbCtxt::Initialize(): Database ID: [6] 'WideWorldImportersDW'. XTP Engine version is 2.9. 2016-11-03 14:56:29.16 spid16s [INFO] HkHostDbCtxt::Initialize(): Database ID: [5] 'WideWorldImporters'. XTP Engine version is 2.9. 2016-11-03 14:56:29.18 spid17s [INFO] HkHostDbCtxt::Initialize(): Database ID: [6] 'WideWorldImportersDW'. XTP Engine version is 2.9. 2016-11-03 14:56:29.26 spid18s [INFO] Database ID: [5]. Deleting unrecoverable checkpoint table row (id: 2). 2016-11-03 14:56:29.30 spid19s [INFO] Database ID: [6]. Deleting unrecoverable checkpoint table row (id: 3). 2016-11-03 14:56:29.45 spid18s [INFO] HkCkptLoadInternalEx(): Database ID: [5]. Root file: {3E231B6B-9EF1-4AE3-A1DB-BD01DC866DD9}, watermark: 40, RecoveryLsn: 00000272:00005EC4:0002, RecoveryCheckpointId: 4, RecoveryCheckpointTimestamp: 0xb 2016-11-03 14:56:29.48 spid19s [INFO] HkCkptLoadInternalEx(): Database ID: [6]. Root file: {01177532-CE1C-4F26-BBFD-08AC9719C58E}, watermark: 160, RecoveryLsn: 00000040:0000017E:0002, RecoveryCheckpointId: 5, RecoveryCheckpointTimestamp: 0x11

SQL Server Tiger Team

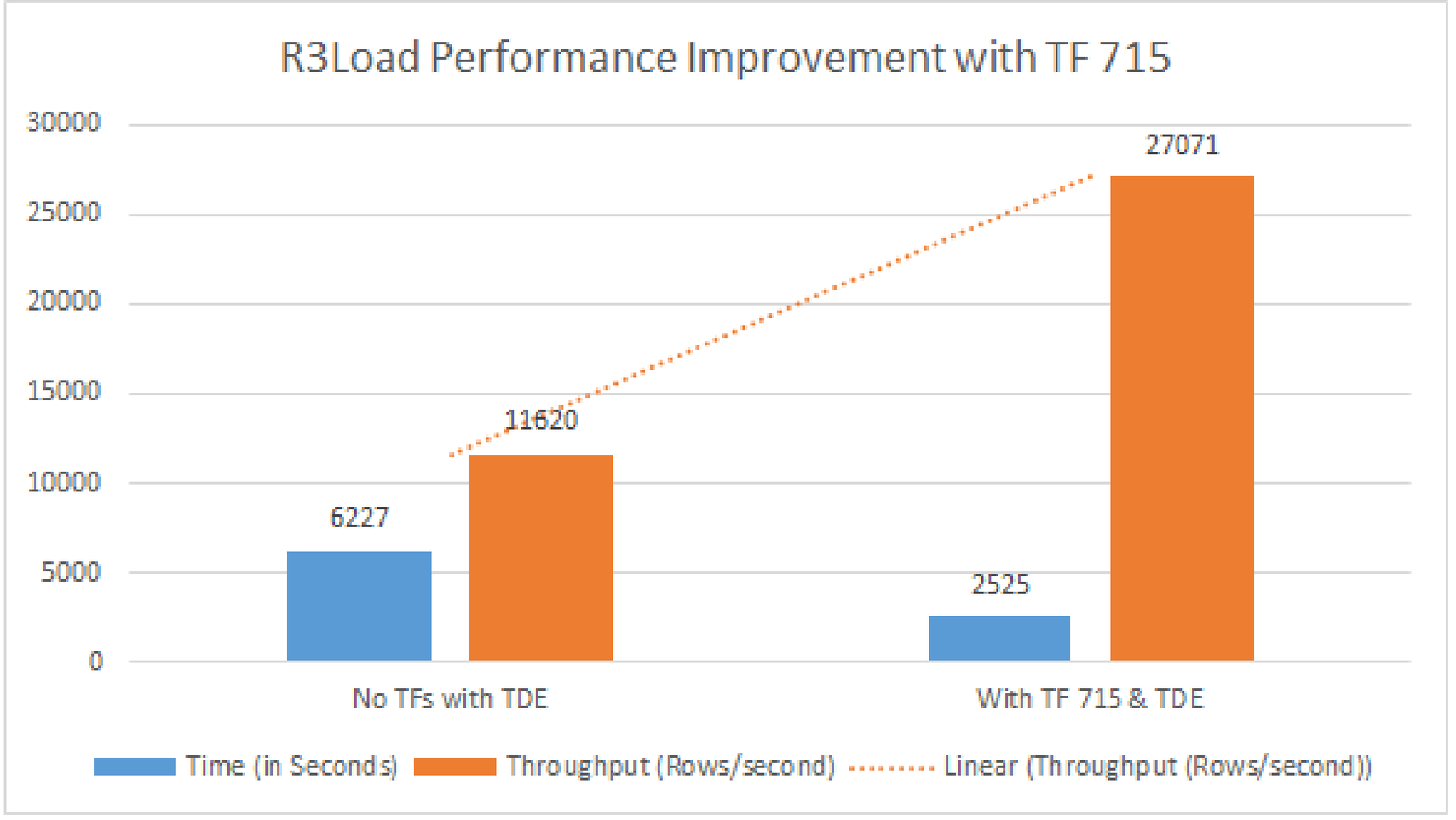
Supportability Improvements

- A new errorlog message logged indicating **number of tempdb** files or **if different tempdb file size** or **autogrow setting** is found.

```
2016-11-16 01:42:25.74 spid4s      Starting up database 'tempdb'.
2016-11-16 01:42:26.11 spid4s      The tempdb database has 4 data file(s).
2016-11-16 01:42:26.11 spid4s      The tempdb database data files are not configured
with the same initial size and autogrowth settings. To reduce potential allocation
contention, the initial size and autogrowth of the files should be same.
```

Performance Enhancements/Changes

- Bulk insert into heaps with AUTO TABLOCK under TF 715.



Performance Enhancements/Changes

- SQL 2016 RTM Behavior INSERT..SELECT
  - For User tables – Parallel INSERT requires TABLOCK hint.
  - For Local temp tables – Parallel INSERT by default.
- SQL 2016 SP1 Behavior INSERT..SELECT
  - For User tables – Parallel INSERT requires TABLOCK hint.
  - For Local temp tables – Parallel INSERT requires TABLOCK hint.
- TF 9495 disables parallelism in insert

SQL Server Tiger Team

Relational

Missing Diagnostics In query plans

- Showplan extended in SQL Server 2016 to support:
  - Expose max memory enabled for the query
  - Memory fractions for optimized nested loop join
  - Information about enabled trace flags
  - Memory grant warning

Missing Diagnostics In query plans

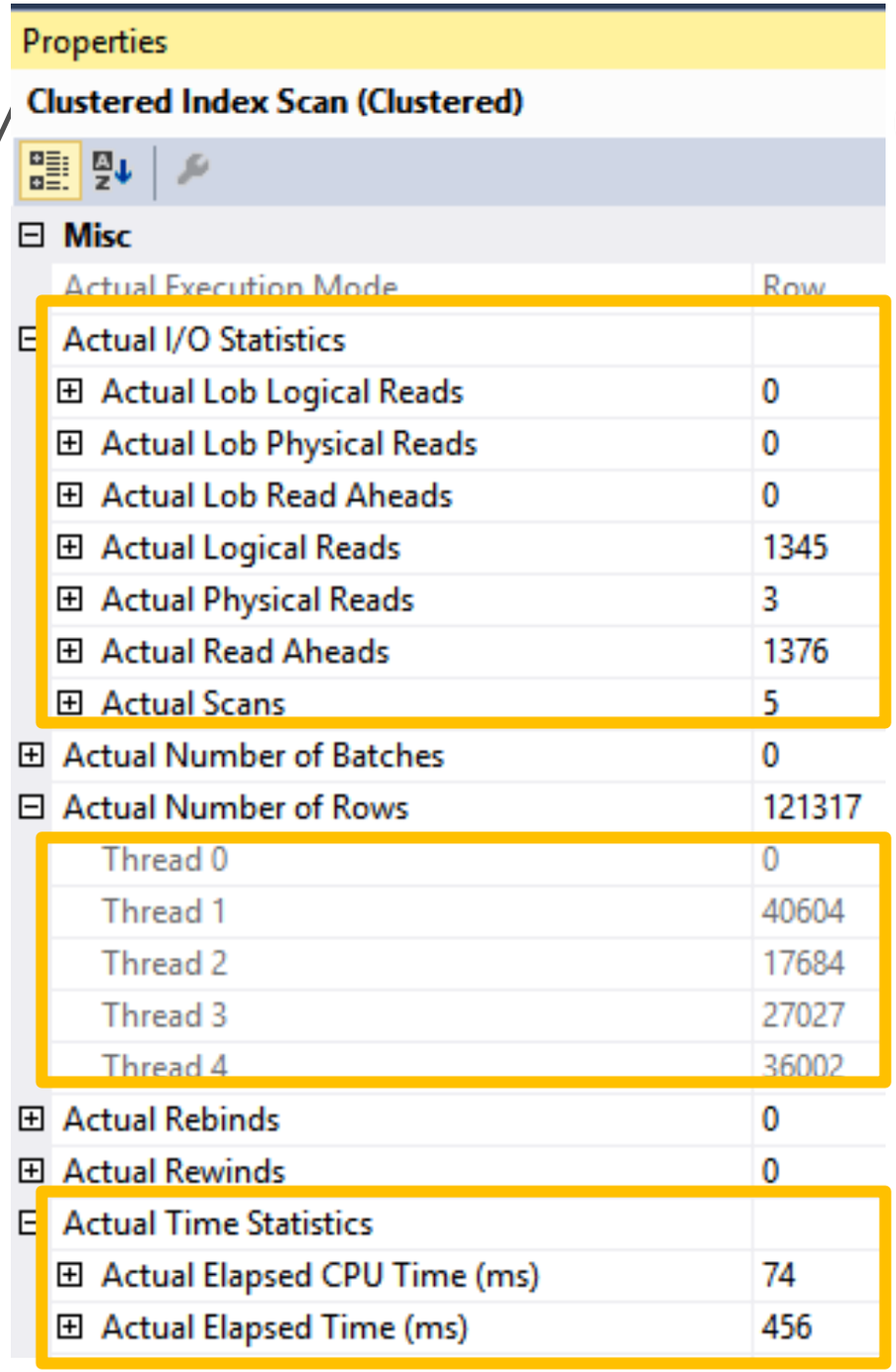
- Showplan extended in SQL Server 2016 SP1 to support:
  - Expose max memory enabled for the query
  - Memory fractions for optimized nested loop join
  - Information about enabled trace flags
  - Memory grant warning
  - **Information about parameters data type**
  - **CPU and execution elapsed time for entire query (root node)**
  - **Top 10 waits (sys.dm_exec_session_wait_stats)**

Missing perf insights on query plan nodes

- Per operator performance statistics for node and threads
- Showplan extended to include *RunTimeCountersPerThread*
- Node costs for parent and children:
  - Cumulative values for Row mode operators
  - Singleton values for Batch mode operators

Runtime Info	Up to SQL 2016
ActualRows	X
ActualRowsRead	
Batches	
ActualEndOfScans	X
ActualExecutions	X
ActualExecutionMode	
ActualElapsedms	
ActualCPUms	
ActualScans	
ActualLogicalReads	
ActualPhysicalReads	
ActualReadAheads	
ActualLobLogicalReads	
ActualLobPhysicalReads	
ActualLobReadAheads	
InputMemoryGrant	
OutputMemoryGrant	
UsedMemoryGrant	

M on query plan nodes




```
<RunTimeInformation>
<RunTimeCountersPerThread Thread="0" ActualRows="121317"
ActualRowsRead="10000000" Batches="0" ActualEndOfScans="3"
ActualExecutions="1" ActualExecutionMode="Row"
ActualElapsedms="456" ActualCPUms="74" ActualScans="3"
ActualLogicalReads="1345" ActualPhysicalReads="3"
ActualReadAheads="1376" ActualLobLogicalReads="0"
ActualLobPhysicalReads="0" ActualLobReadAheads="0" />
</RunTimeInformation>
```

Per-operator level performance stats

- New xEvent query_thread_profile in SQL Server 2016
  - Showplan time scale = milliseconds
  - xEvent time scale = microseconds for CPU and total time.

Name	Category	Channel
query_thread_profile	execution	Debug

query_thread_profile

Reports the performance of each node and thread of a query plan after execution

Event Fields	Description
actual_batches	Number of batches processed by this thread
actual_execution_mode	Execution mode of the thread. 0 indicates row mode, 1 indicates batch mode
actual_logical_reads	Number of logical pages read
actual_physical_reads	Number of physical pages read
actual_ra_reads	Number of read-ahead pages read
actual_rebinds	Number of rebinds for this thread
actual_rewinds	Number of rewinds for this thread
actual_rows	Number of rows processed by this thread
actual_writes	Number of pages written
cpu_time_us	CPU time in microseconds
io_reported	Is IO reported?
node_id	The ID of the node in the query plan
thread_id	The ID of the thread running in this node
total_time_us	Cumulative time in microseconds, including waits

Detecting predicate search inefficiencies?

- *Actual number of rows* returned are rows after the predicate is applied.
  - Not the actual number of rows that are scanned from a table or index.
- Scenario hidden from an actual execution plan:
  - SCAN or SEEK returns only 10 rows, why is it taking so long?
  - You see high CPU or many logical reads, but the query plan doesn't reflect that.
- Now what?? ☹

Predicate Pushdown as seen in Showplan

SQL Server Tiger Team

```
SELECT * FROM [Production].[TransactionHistory]
WHERE [ProductID] = 870 AND [Quantity] > 10
```

Key Stats

- *Actual Rows in result set = 39*
- *Actual Rows Read = 113443*

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	Row
Estimated Execution Mode	Row
Storage	RowStore
Actual Number of Rows	39
Actual Number of Batches	0
Estimated Operator Cost	0.651523 (88%)
Estimated I/O Cost	0.589051
Estimated CPU Cost	0.0624722
Estimated Subtree Cost	0.651523
Number of Executions	4
Estimated Number of Executions	1
Estimated Number of Rows	1500.73
Estimated Row Size	54 B
Actual Rebinds	0
Actual Rewinds	0
Ordered	False
Node ID	1

SQL 2014

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	Row
Estimated Execution Mode	Row
Storage	RowStore
Number of Rows Read	113443
Actual Number of Rows	39
Actual Number of Batches	0
Estimated I/O Cost	0.589051
Estimated Operator Cost	0.620287 (92%)
Estimated Subtree Cost	0.620287
Estimated CPU Cost	0.0312361
Number of Executions	8
Estimated Number of Executions	1
Estimated Number of Rows	1500.73
Estimated Row Size	54 B
Actual Rebinds	0
Actual Rewinds	0
Ordered	False
Node ID	1

SQL 2016

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	Row
Estimated Execution Mode	Row
Storage	RowStore
Number of Rows Read	113443
Actual Number of Rows	39
Actual Number of Batches	0
Estimated I/O Cost	0.589792
Estimated Operator Cost	0.621028 (92%)
Estimated CPU Cost	0.0312361
Estimated Subtree Cost	0.621028
Number of Executions	8
Estimated Number of Executions	1
Estimated Number of Rows	1500.73
Estimated Number of Rows to be Read	113443
Estimated Row Size	54 B
Actual Rebinds	0
Actual Rewinds	0
Ordered	False
Node ID	1

SQL 2016 SP1

Influencing query execution without sysadmin

```
SELECT AddressID
FROM Person.[Address]
WHERE City = N'Ballard'
        AND [PostalCode] = '98107'
OPTION (QUERYTRACEON 9481)
```

```
SELECT AddressID
FROM Person.[Address]
WHERE City = N'Ballard'
        AND [PostalCode] = '98107'
OPTION (USE HINT('FORCE_LEGACY_CARDINALITY_ESTIMATION'))
```

- USE HINT option does not require sysadmin privileges.
- 9 different hints are supported to enable functionality which was previously only available with trace flags.
- Use **sys.dm_exec_valid_use_hints** DMV to see the list of all supported hints under the USE HINT notation.

Available USE HINT query hints

USE HINT	TF ?	DB Option ?
DISABLE_OPTIMIZED_NESTED_LOOP	2340	
FORCE_LEGACY_CARDINALITY_ESTIMATION	9481	Yes
ENABLE_QUERY_OPTIMIZER_HOTFIXES	4199	Yes
DISABLE_PARAMETER_SNIFFING	4136	Yes
ASSUME_MIN_SELECTIVITY_FOR_FILTER_ESTIMATES	4137 for OldCE 9471 for NewCE	
DISABLE_OPTIMIZER_ROWGOAL	4138	
ENABLE_HIST_AMENDMENT_FOR_ASC_KEYS	4139	
ASSUME_JOIN_PREDICATE_DEPENDS_ON_FILTERS	9476 on NewCE	
FORCE_DEFAULT_CARDINALITY_ESTIMATION	2312	

aka.ms/traceflags

USE HINT

DEMO

CREATE OR ALTER

- Increase Developer productivity.
- CREATE OR ALTER support to make it easier to modify and deploy objects like Stored Procedures, Triggers, UDFs and Views.
  - `CREATE OR ALTER PROCEDURE <…>`
- Why not Tables and Schemas?
  - Syntax for CREATE and ALTER related to TABLE or SCHEMA are two very different commands from a syntax and usability perspective.

```
CREATE TABLE T3
(C1 int PRIMARY KEY,
C2 varchar(50) NULL,
C3 int NULL,
C4 int ) ;
GO
```

```
ALTER TABLE T3
ALTER COLUMN C2 varchar(50) COLLATE
Latin1_General_BIN;
GO
```

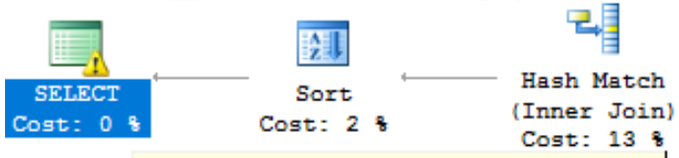
SQL Server Tiger Team

New Memory Grant Showplan Warning

SQL Server 2014 SP2 and SQL Server 2016 SP1

- 3 conditions:
  - **Excessive Grant**: when max used memory is too small compared to the granted memory. This scenario can cause blocking and less efficient usage when large grants exist and a fraction of that memory was used.

KB 3172997



SELECT

Actual Number of Rows	0
Cached plan size	64 KB
Degree of Parallelism	0
Estimated Operator Cost	0 (0%)
Estimated Subtree Cost	0.205452
Memory Grant	67808
Estimated Number of Rows	89.3525

Statement

```
SELECT
[fo].[Order Key], [fo].[Description]
FROM    [Fact].[Order] AS [fo]
INNER HASH JOIN [Dimension].[Stock
Item] AS [si]
ON [fo].[Stock Item Key] = [si].[Stock Item
Key]
WHERE   [fo].[Lineage Key] =
@LineageKey
AND [si].[Lead Time Days] > 0
ORDER BY [fo].[Stock Item Key], [fo].[Order
Date Key] DESC
OPTION (MAXDOP 1)
```

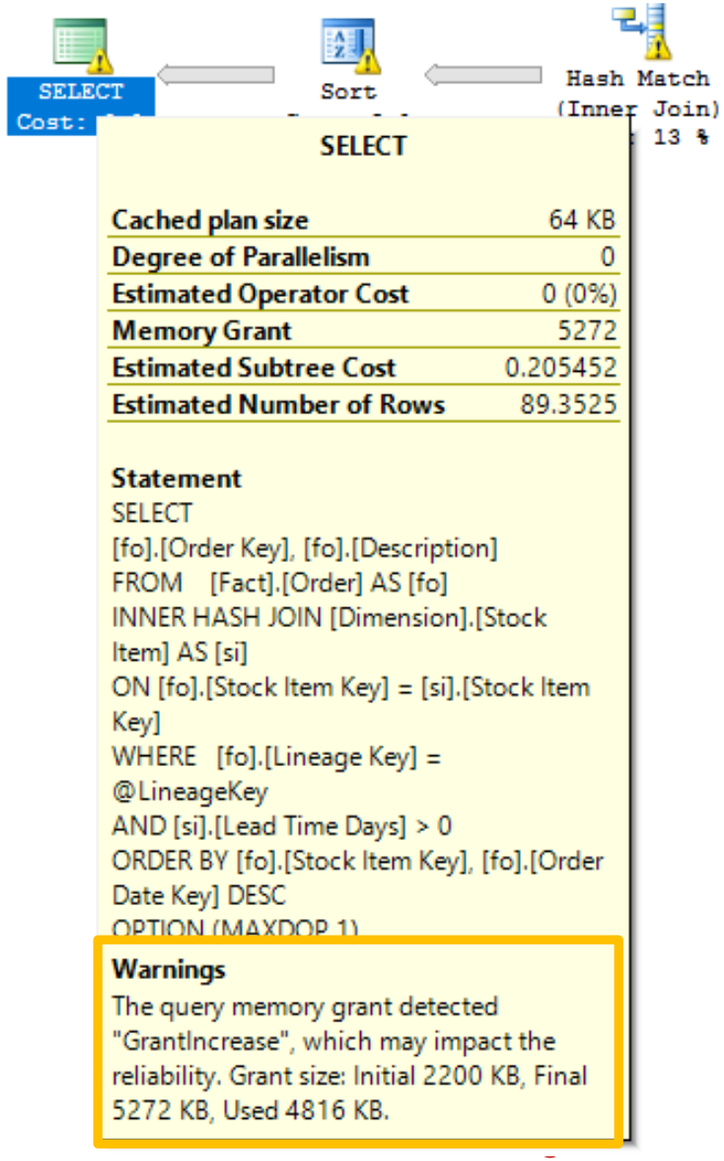
Warnings

The query memory grant detected "ExcessiveGrant", which may impact the reliability. Grant size: Initial 67808 KB, Final 67808 KB, Used 1024 KB.

New Memory Grant Showplan Warning

SQL Server 2014 SP2 and SQL Server 2016 SP1

- 3 conditions:
  - **Excessive Grant**: when max used memory is too small compared to the granted memory. This scenario can cause blocking and less efficient usage when large grants exist and a fraction of that memory was used.
  - **Grant Increase**: when the dynamic grant starts to increase too much, based on the ratio between the max used memory and initial request memory. This scenario can cause server instability and unpredictable workload performance.
  - **Used More Than Granted**: when the max used memory exceeds the granted memory. This scenario can cause OOM conditions on the server.



Diagnostics

Demo

Tracking query progress (estimated)

- To have in-flight query execution statistics, the query execution statistics profile infrastructure must be enabled on demand.
- Can be enabled for a target session:
  - Specifying Include Live Query Statistics in SSMS.
  - SET STATISTICS XML ON
  - SET STATISTICS PROFILE ON

- Or globally to view the LQS from other sessions (such as from Activity Monitor):
  - Enabling *query_post_execution_showplan* extended event.
- High overhead (75% with TPC-C like workload)

Lightweight Tracking query progress (estimated)

- Lightweight query execution profiling dramatically reduces performance overhead of continuously collecting per-operator query execution statistics (such as actual number of rows)
- Can be enabled by:
  - Using global TF 7412.
  - Enabling *query_thread_profile* extended event.
  - When lightweight profiling is on, *sys.dm_exec_query_profiles* is also populated for all sessions.

- This enables usage of LQS feature in SSMS (including Activity Monitor) and on the new DMF sys.dm_exec_query_statistics_xml.
- The following still use regular profiling infra:
  - SET STATISTICS XML (or Include Actual Plan).
  - *query_post_execution_showplan* extended event.

What is the impact of live query troubleshooting?

Query Execution Statistics Profiling Infrastructure tests with TPC-C like workloads

	Overhead percent (up to)	
Infra Type	**no active xEvents**	**Active xEvent query_post_execution_showplan**
Legacy	**75.5**	93.17
Lightweight in SQL Server 2014 SP2/2016	3.5	62.02
Lightweight in SQL Server 2016 SP1	**2**	14.3

What Diagnostic choices do you have?

SQL Server Tiger Team

Regular Profiling

- Full runtime statistics for a query plan
- Most expensive overhead
- Can be enabled per session or globally
- Consume data from live queries or post execution

Lightweight Profiling

- Limited runtime statistics in query plan (no CPU tracking)
- Least expensive overhead
- Only enabled globally
- Consume data from live queries or post execution
- More optimized in SQL Server 2016 SP1

Query Progress

DEMO

Miscellaneous

KB 3191296

- Track TLS protocol with trace xEvent.

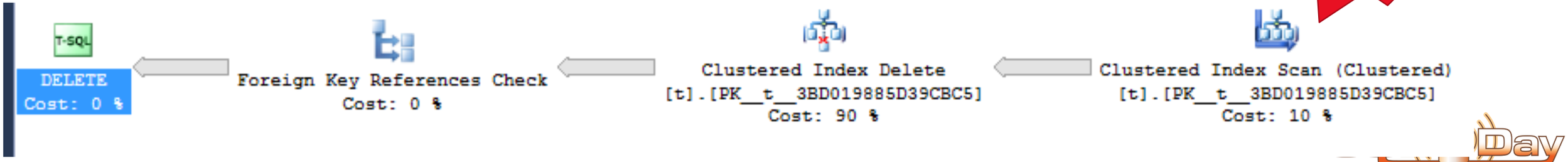
trace	2016-11-10 14:20:07.1448972	ssl.cpp	Ssl::Handshake	6853	SNISecurity Handshake Handshake succeeded. Protocol: TLS1.2 (1024), Cipher: AES 256 (26128), Cipher Strength: 256

KB 3170114

- sys.dm_db_incremental_stats_properties contains a partition number column, otherwise similar to sys.dm_db_stats_properties for non-incremental statistics.

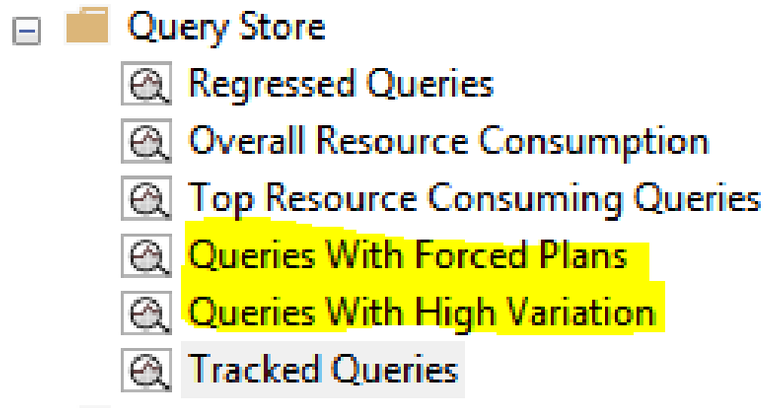
KB 3191273

- Referential Integrity Operator will also support self-referential constraints.



Query performance insights in SSMS

- Still in last v16:
  - Support for multi-statement showplan comparison
  - Per-operator level performance stats in showplan Properties window
  - Query Store
    - Filter by number of different plans



- New with v17:
  - Query Store: new reports
  - Query analysis scenarios: Introduces CE diff search
  - Statistics loaded by the optimizer (SQL Server 2017)

What's next for Relational?

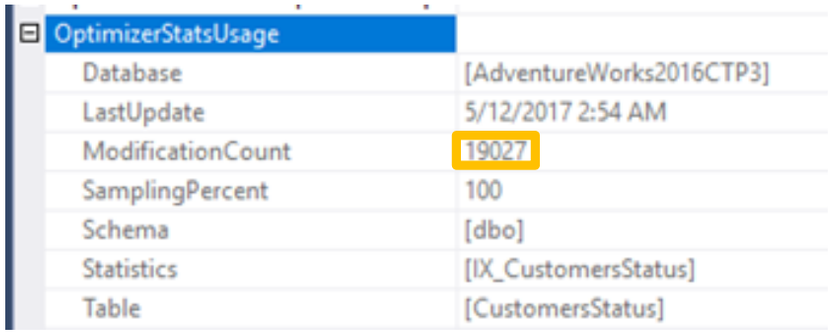
- SQL Server 2017
  - Automatic Query Plan Tuning
  - Adaptive QP
    - Mem Grant Feedback; Interleaved Execution; Adaptive JOINs
- SQL Server 2012 SP4
  - **Extended diagnostics in showplan XML**: information about enabled trace flags, memory fractions for optimized nested loop join, CPU time and elapsed time
  - Better **correlation** between diagnostics XE and DMVs
  - Better **memory grant/usage diagnostics**

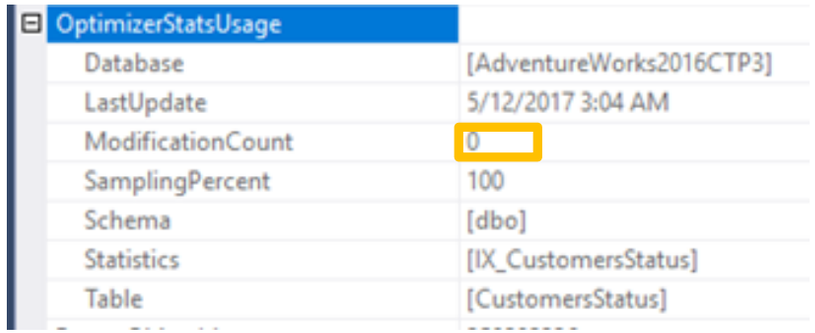
SQL Server Tiger Team

Community Driven Improvements in SQL Server 2017

Statistics information in Showplan

- Identify which statistics were used by the Query Optimizer for a given compilation.
- Gain actionable insight to where estimations came from.



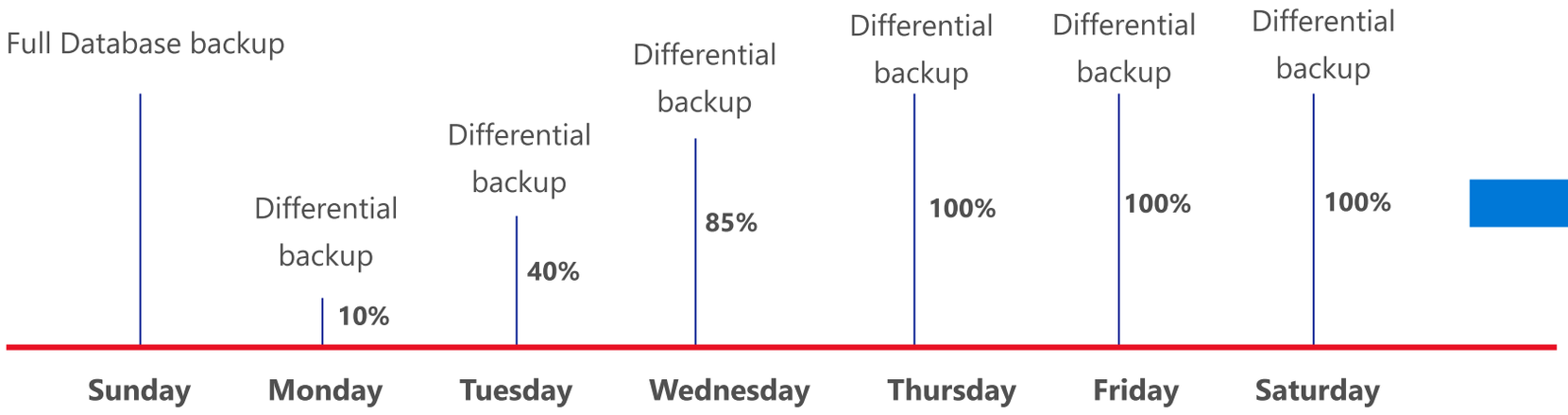


Statistics information in Showplan

DEMO

Smart Differential Backup

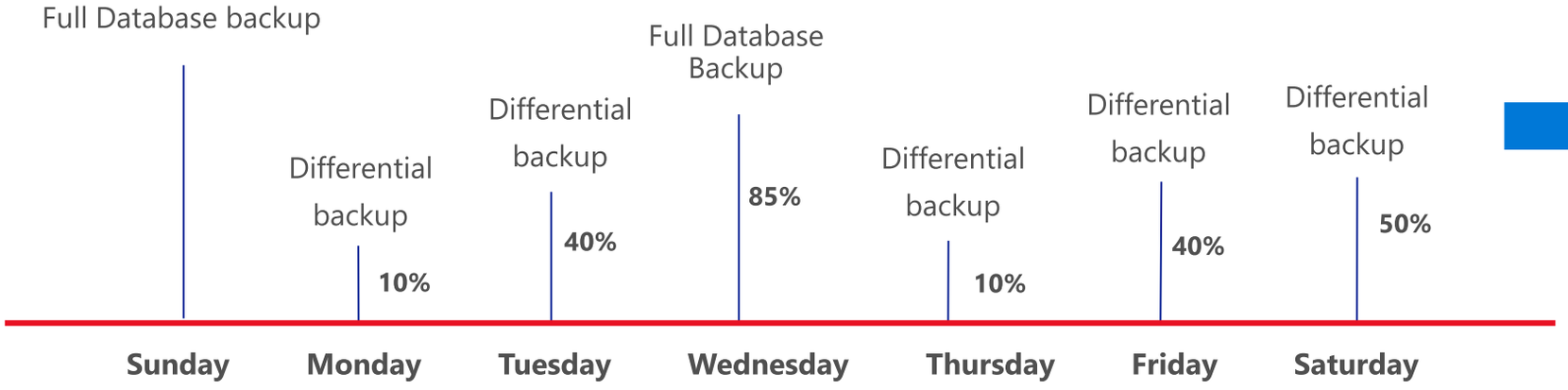
Current Differential Backup weekly cycle example



Restore chain too long impacting RTO.

Making Differential Backup smarter

modified_extent_page_count -> sys.dm_db_file_space_usage

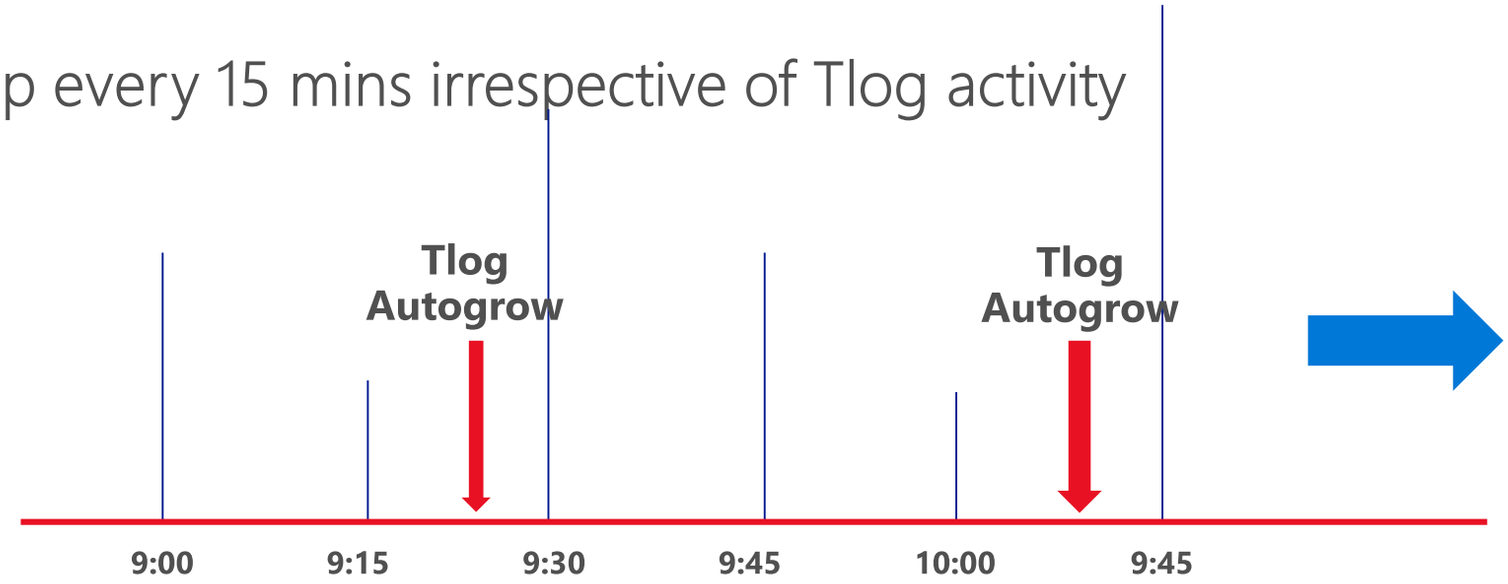


Faster restore
Storage Savings

SQL Server Tiger Team

Smart Transaction log backup

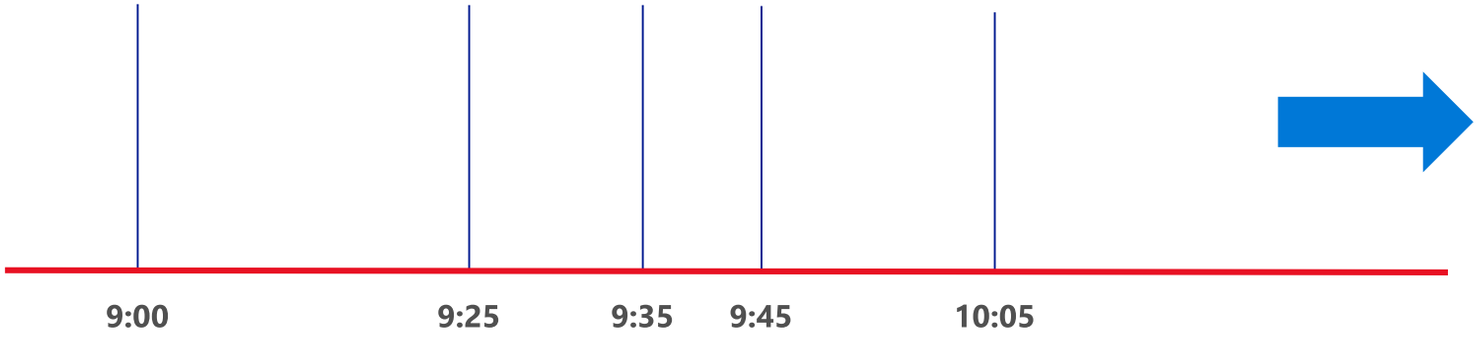
Backup every 15 mins irrespective of Tlog activity



Unpredictable
- Tlog Activity
- Size of tlog backup
- backup timings
- Autogrows

Smart Transaction Log Backup

log _since_last_log_backup_mb -> sys.dm_db_log_stats(database_id)



Consistent
- Tlog backup size
- backup timings
- No Autogrow

Smart Diff Backup

DEMO

SELECT INTO..ON Filegroup

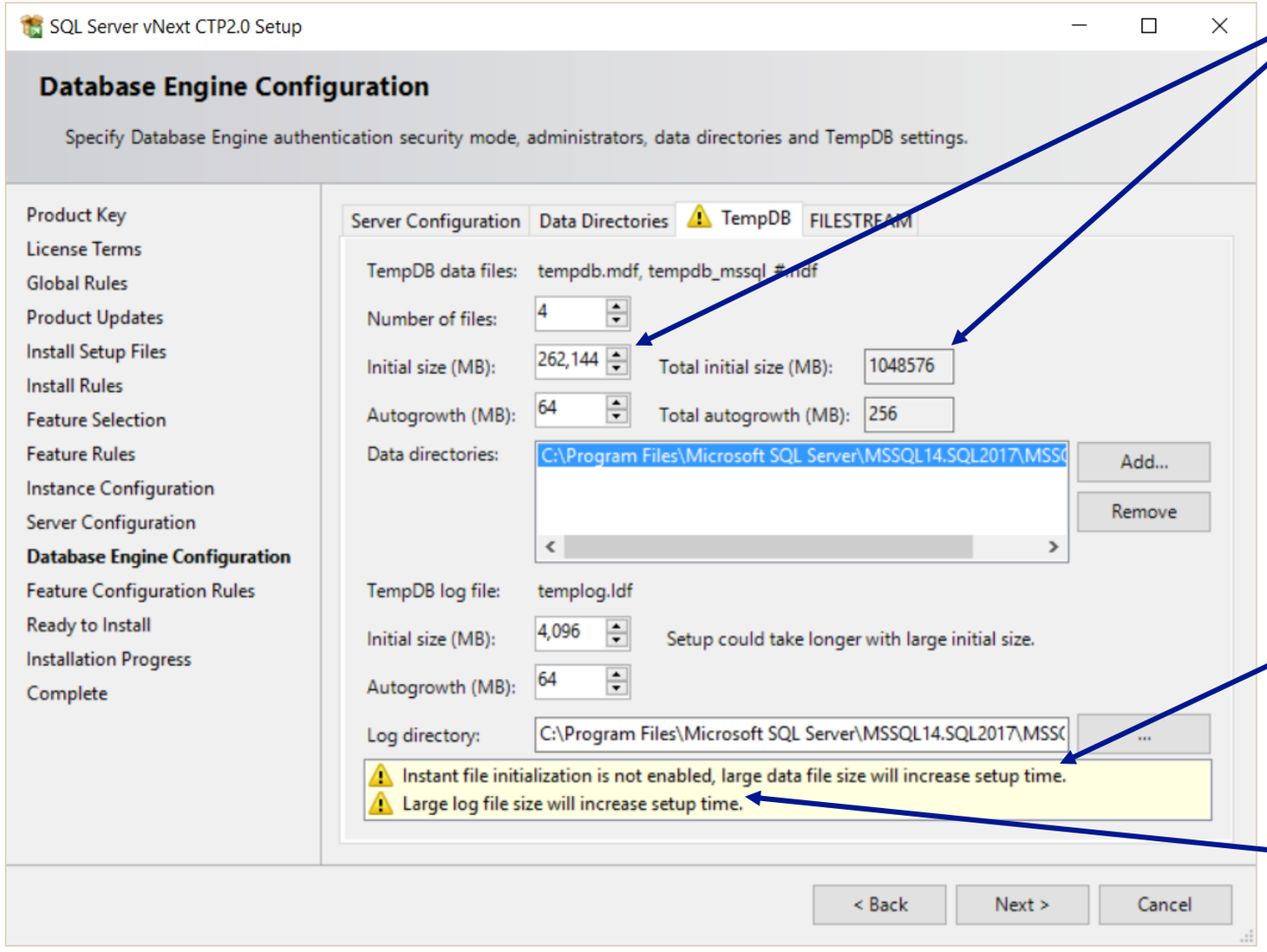
```
ALTER DATABASE [AdventureWorksDW2016] ADD FILEGROUP FG2

SELECT * FROM sys.database_files

ALTER DATABASE [AdventureWorksDW2016]
ADD FILE
(
NAME='FG2_Data',
FILENAME = '/var/opt/mssql/data/AdventureWorksDW2016_Data1.mdf'
)
TO FILEGROUP FG2;
GO

SELECT *  INTO [dbo].[FactResellerSalesXL] ON FG2
FROM [dbo].[FactResellerSales];
```

TempDB setup Improvements



Max Tempdb file size uplifted to 256GB (262,144 MB)

Large values increases setup time

Warning if IFI is not enabled and initial size > 1GB

Warning if Log file size > 1GB irrespective of IFI setting

SQL Server Tiger Team

TempDB version store usage and planning

- New DMV sys.dm_tran_version_store_space_usage
- Performant with no overheads (as compared to sys.dm_db_session_space_usage)
- Useful for TempDB space planning

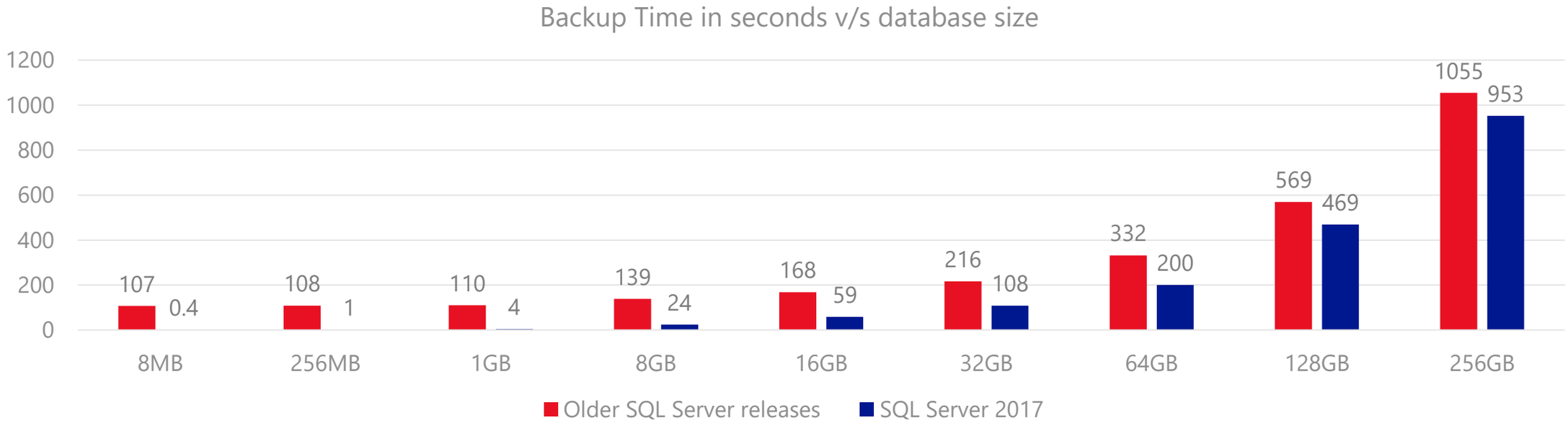
```
SELECT
  DB_NAME(database_id) as 'Database Name',
  reserved_page_count,
  reserved_space_kb
FROM sys.dm_tran_version_store_space_usage;
```

```
Database Name            reserved_page_count reserved_space_kb
------------------------ ------------------- -----------
msdb                      0                   0
AdventureWorks2016        10                  80
AdventureWorks2016DW      0                   0
WideWorldImporters        20                  160
```

Tlog Monitoring and Diagnostics

- DBCC LOGINFO -> sys.dm_db_log_info
- New DMF sys.dm_db_log_stats(dbid)
  - Min LSN
  - Log truncation holdup reason (log reuse wait type desc)
  - VLF Information (vlf_count, active_vlfs, inactive_vlfs)
  - Backup information (last tlog backup information across secondary replicas, backup lsn, log accumulated since last backup)
  - Checkpoint information (last checkpoint lsn, log accumulated since last checkpoint)
  - Recovery information (recovery_lsn, log to recover)

Backup Improvements



DB Size	Previous SQL Server releases	SQL Server 2017	Improvement
8MB	107	0.4	642x
256MB	108	1	108x
1GB	110	4	27.5x
8GB	139	24	5.79x
16GB	168	59	2.85x
32GB	216	108	2.12x
64GB	332	200	66%
128GB	569	469	21.32%
256GB	1055	953	10.70%

SQL Server Tiger Team

Other Improvements

- Processor Information in sys.dm_os_sys_info
  - socket_count
  - cores_per_socket
  - numa_node_count
- DBCC CLONEDATABASE improvements
  - Flush Query Store runtime statistics while creating clone.
  - Support for fulltext indexes.

Bookmarks

SQL Server Tiger Team Blog	http://aka.ms/sqlserverteam
Tiger Toolbox GitHub	http://aka.ms/tigertoolbox
SQL Server Release Blog	http://aka.ms/sqlreleases
BP Check	http://aka.ms/bpcheck
SQL Server Standards Support	http://aka.ms/sqlstandards
Trace Flags	http://aka.ms/traceflags
SQL Server Support lifecycle	http://aka.ms/sqllifecycle
SQL Server Updates	http://aka.ms/sqlupdates
Twitter	@mssqltiger

Polish SQL Server User Group

Data Community

SQL Day 2017

GOLD SPONSORS

Atos

Profisee®

Zerto

SILVER SPONSORS

PROMISE
GRUPA APN PROMISE

ELITMIND

ITMAGINATION
Imagine it. Done.

THEOBALD SOFTWARE

BRONZE SPONSORS

it KONTRAKT®

JCommerce

IT
Krajowy Rejestr Długów

KSIĄŻKI IT Z CAŁEGO ŚWIATA
NOVATECH
www.novatech.com.pl

PEPCO

dbWatch
DATABASE CONTROL

querona

Data Scientist as a Service

UBS

VOLVO
Volvo Group IT

STRATEGIC PARTNER

Microsoft