Gianluca Hotz

SQL Server Database Engine 2017 Enhancements


PASS


thinkIT


community


CONSORZIO
UNIVERSITARIO
DI PORDENONE
MOLTIPLICATORE DI VALORE


Microsoft


vmware


Datamaze


TIMEXTENDER


Real Comm
Engineering IT makes it easy


beanTech
IT moves your business


Quest


PASS
SQLSATURDAY
PORDENONE | 17 FEB 2018

Who am I

Gianluca Hotz | @glhotz | ghotz@ugiss.org

Fondatore e Mentor SolidQ

- 20+ anni con SQL Server (dalla 4.21 nel 1996)
- Modellazione basi di dati, dimensionamento e amministrazione, sviluppo, ottimizzazione

Interessi

- Modello relazionale, architettura DBMS, alta disponibilità e Disaster Recovery

Community

- 20 anni Microsoft MVP SQL Server (dal 1998)
- Fondatore e presidente UGISS
  - User Group Italiano SQL Server (PASS Chapter)

Agenda

- Configuration
- Development — Demo First!
- In-Memory
- Administration — Demo First!
- HA & DR
- Performance
- Q&A

Configuration

Installation

Separate downloads

- SQL Server Management Studio 17.x (use it also for SQL 2016)
- SQL Server Data Tools
- **New in 2017: Reporting Services**

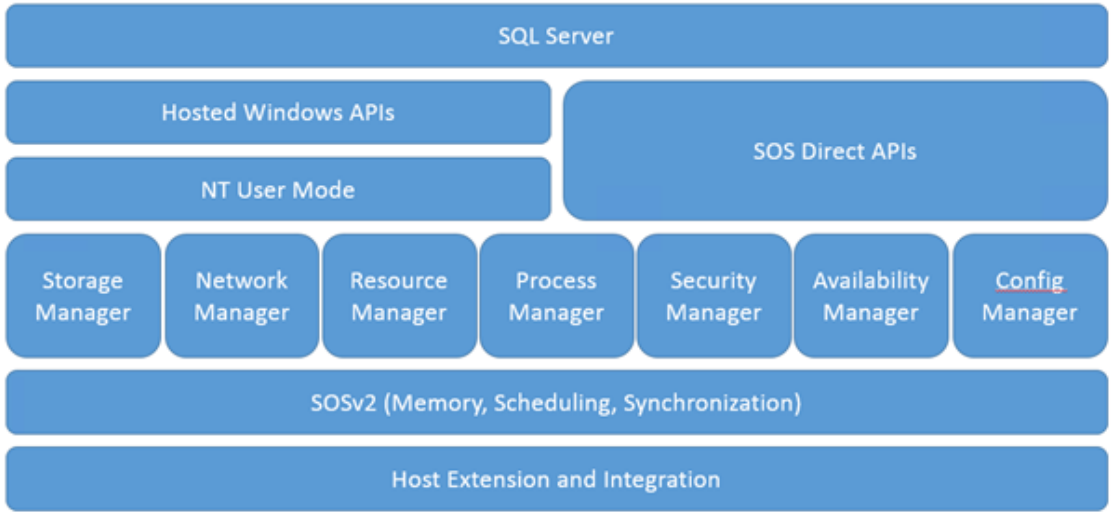
Tempdb configuration

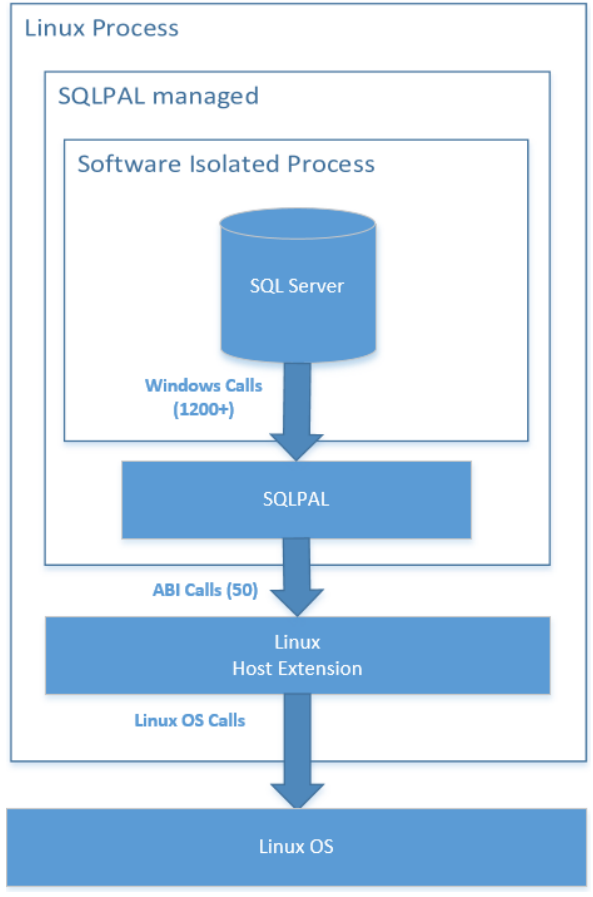
- Allows up to 256GB file size
- Warning if > 1GB and Instant File Initialization not enabled

Linux

- Easy: import repo GPG keys, register repository, apt-get update/install
- **mssql-conf** to script out installation (or to be used in a script installation)
- can use environment variables

SQL on Linux Architecture





SQL on Linux Supported Distro/Versions

Red Hat Enterprise Linux 7.3 or 7.4

Ubuntu Linux 16.04 LTS

Suse Linux Enterprise Server v12 SP2

Docker Engine 1.8+ on Linux, Mac, Windows

SQL on Linux Minimum Requirements

2 x core CPU 2Ghz (x64 only)

3,25 GB memory

6GB disk space (XFS or EXT4 file systems)

SQL on Linux Packages

mssql-tools

ODBC libraries
sqlcmd command-line querying tool
bcp for bulk import/export

mssql-server-agent

mssql-server-fts

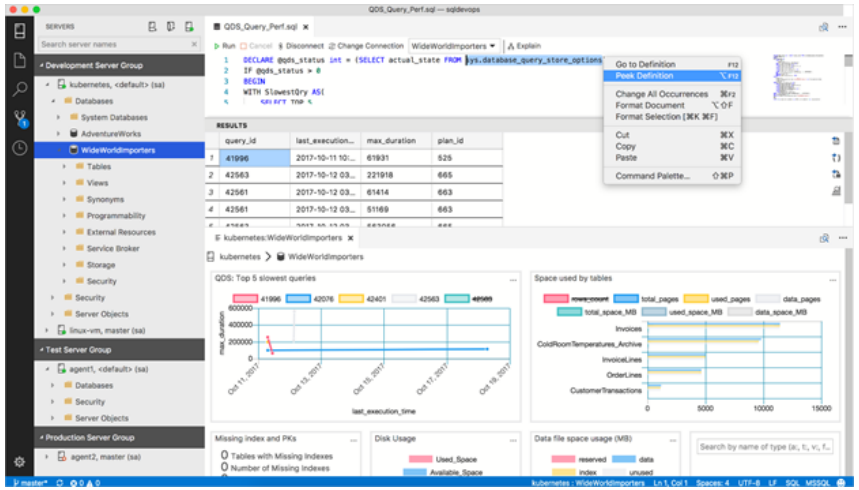
Full-Text indexing and Semantic Search (requires additional steps)

mssql-server-is

Tools

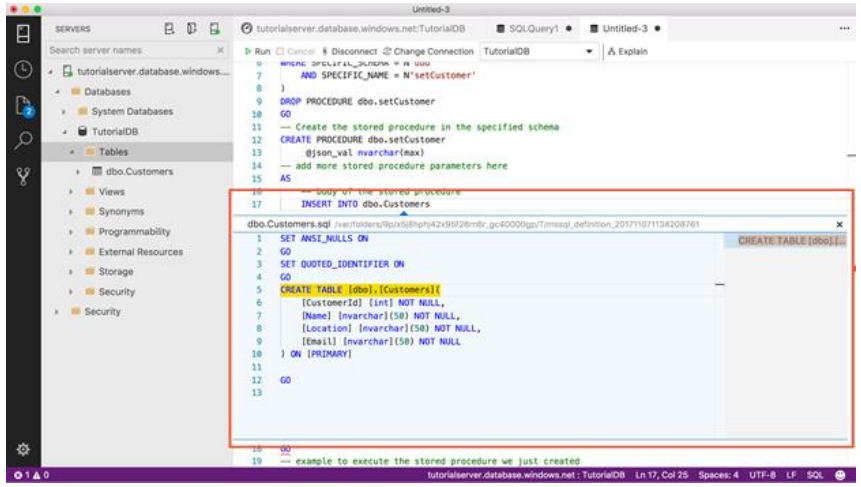
In Linux

- Configuration script **mssql-conf**
- T-SQL commands with **sqlcmd**



In Windows

- SQL Server Management Studio (SSMS)
- Visual Studio with SQL Server Data Tools (SSDT)
- Visual Studio Code with **mssql** extension
- PowerShell
- T-SQL commands with **sqlcmd**



Multi platform (Windows, Mac, Linux)

- SQL Operation Studio (preview)

CLR Strict Security

Code Access Security (CAS) in.NET Framework

- No more supported as security boundary
- Assembly with **PERMISSION_SET = SAFE** may access external resources, call unmanaged code, acquire sysadmin privileges

New configuration option “CLR Strict Security”

- Treats **SAFE/EXTERNAL ACCESS** as **UNSAFE**
- Requires assemblies to be signed and login with **UNSAFE ASSEMBLY** permission
  (database with TRUSTWORTHY ON owned by login with UNSAFE ASSEMBLY permission)
- Enabled by default can be disabled for backward compatibility (not recommended)

White-listing

- **sp_add_trusted_assembly**, **sp_drop_trusted_assembly**

Database Configuration

ALTER DATABASE SCOPED CONFIGURATION

- Clear procedure cache
- Cardinality estimation independent of compatibility level
- Parameter sniffing
- Query optimization hotfixes
- **New in 2017**: identity cache, turn off to avoid gaps (like Trace Flag 272)

DATABASE SCOPED CREDENTIAL

- Now a securable class supporting CONTROL, ALTER, REFERENCES, TAKE OWNERSHIP, VIEW DEFINITION

Development

Demo First!

Database Development

SELECT...INTO

Supports **ON** keyword to specify filegroup

New bulk access options

FORMAT = 'CSV'

Support for RFC 4180 in BULK INSERT and OPENROWSET

CREATE EXTERNAL DATA SOURCE ... TYPE = BLOB_STORAGE

Point to specific LOCATION URL

Use Shared Access Signatures (SAS) via CREDENTIAL

Temporal Tables

Support for **CASCADE DELETE** and **CASCADE UPDATE**

Retention Policy support

- Instead of custom cleanup script, stretch database or table partitioning
- Enabled at database level
  - ALTER DATABASE SET … **TEMPORAL_HISTORY_RETENTION ON**
- Specified per table
  - e.g. WITH(SYSTEM_VERSIONING = ON(… **HISTORY_RETENTION_PERIOD = 6 MONTHS**)
  - Support for DAYS, WEEKS, MONTHS, YEARS and INFINITE
  - Checked against column representing end of SYSTEM_TIME period

T-SQL Development

CONCAT_WS

Concatenate strings (columns/values) using first one as separator

TRANSLATE

Replaces set of characters in string with other set of character
e.g. **SELECT TRANSLATE('2*[3+4]/{7-2}', '[]{}', '()()');**

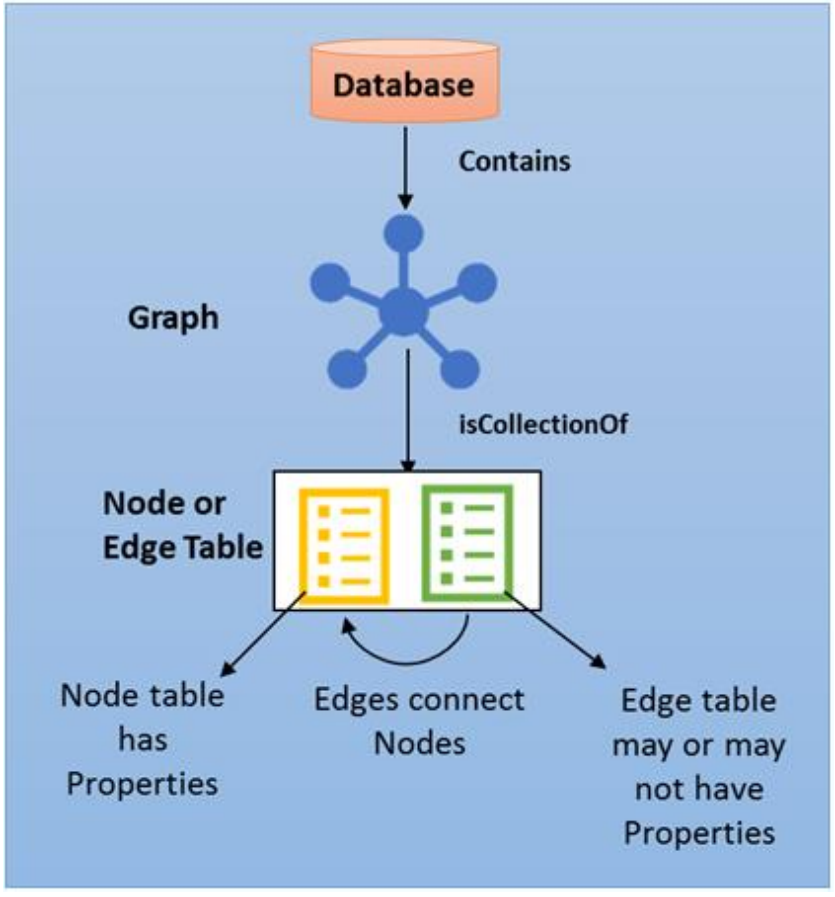
TRIM

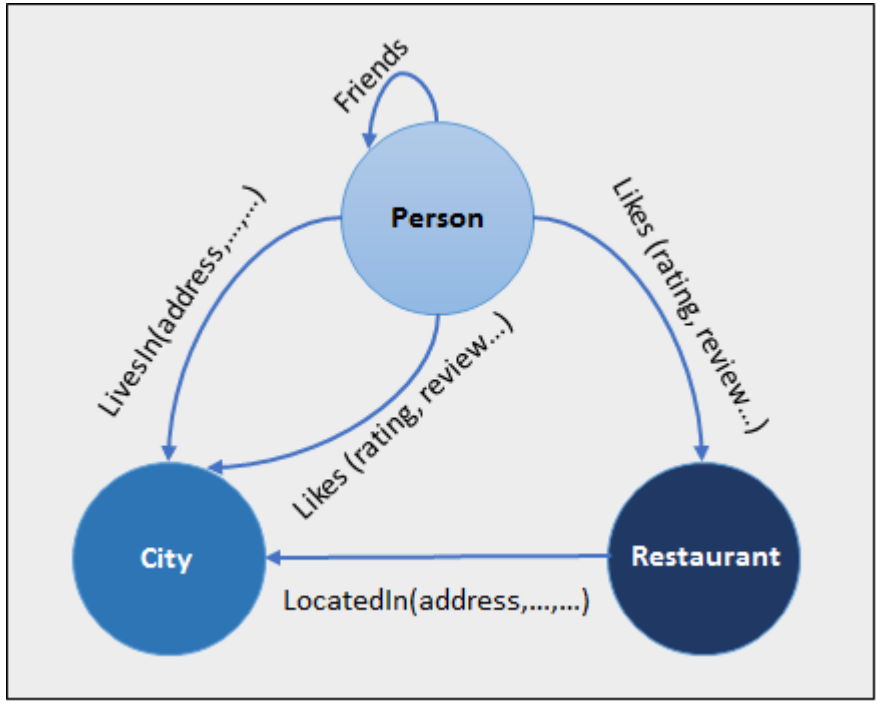
Finally!!!! ☺
Can trim a set of characters

STRING_AGG

Concatenates strings (rows) using separator

Graph Processing





Graph Processing Concepts

Node Table

Represent an entity

Edge Table

Represent many-to-many relationship
May have properties
Is directed, connects two nodes

MATCH T-SQL Command

Search condition for graph objects
Pattern matching and traversal

In-Memory

Columnstore Indexes in SQL Server 2017

Support for non-persisted computed column

Support for LOBs

- Clustered Columnstore only
- Already available also in Azure SQL Database (Premium)
- E.g. JSON data case
  - https://blogs.msdn.microsoft.com/sqlserverstorageengine/2017/02/09/json-data-in-clustered-column-store-indexes

Online non-clustered Columnstore index build/rebuild support

Feature summary for product releases

- https://docs.microsoft.com/en-us/sql/relational-databases/indexes/columnstore-indexes-what-s-new

Memory-Optimized Tables in SQL 2017

Support for computed columns

- Including indexes on computed columns

Support for **JSON** functions

- Both natively compiled T-SQL modules and check constraints

Increased support in natively compiled T-SQL modules

- **CROSS APPLY, CASE, TOP (N) WITH TIES, sp_spaceused, sp_rename**

Azure Storage support for memory-optimized files

- Including backup/restore

Performance

- Faster ALTER TABLE and bw-tree index rebuild during recovery
- Transaction-log REDO done in parallel

Administration

Demo First!

System Metadata

sys.dm_os_sys_info

Added **socket_count, cores_per_socket, numa_node_count**
Useful with VM

sys.dm_os_host_info

OS information on Windows/Linux

Undocumented

sys.dm_os_enumerate_fixed_drives
sys.dm_os_enumerate_filesystem(dir, pattern)
sys.dm_os_file_exists(dir_or_file)

Database Metadata

sys.dm_db_log_stats()

Summary info about t-log, useful for monitoring

sys.dm_db_file_space_usage

Added **modified_extent_page_count**
Useful to build smart backup solutions eg. differential if < 70-80%, full otherwise

sys.dm_tran_version_store_space_usage

Reserved space in pages/KB

sys.dm_db_stats_histogram(object, stat)

SQL Server 2017 / SQL Server 2016 SP1 CU2

DBCC CLONEDATABASE

Creates empty copy of database for troubleshooting

- No data but full schema
  - (constraints, type, t-sql modules, etc.)
- Statistics
- Non blocking
- Read only by default but can be changed
- Optionally **NO_STATISTICS**, **NO_QUERYSTORE**
- SQL Server 2012 SP4, 2014 SP2 CU3, 2016 SP1, 2017

USE HINT Query Option

Named hints instead of hard to remember trace flags

OPTION (USE HINT('DISABLE_PARAMETER_SNIFFING')) instead of **OPTION (QUERYTRACEON 4136)**

Doesn't require sysadmin permission!

Available in SQL Server 2017 / SQL Server 2016 SP1

List from **sys.dm_exec_valid_use_hints**

ASSUME_JOIN_PREDICATE_DEPENDS_ON_FILTERS	9476
ASSUME_MIN_SELECTIVITY_FOR_FILTER_ESTIMATES	4137, 9471
DISABLE_PARAMETER_SNIFFING	4136
DISABLE_OPTIMIZER_ROWGOAL	4138
DISABLE_OPTIMIZED_NESTED_LOOP	2340
ENABLE_HIST_AMENDMENT_FOR_ASC_KEYS	2389
ENABLE_QUERY_OPTIMIZER_HOTFIXES	4199
FORCE_DEFAULT_CARDINALITY_ESTIMATION	2312
FORCE_LEGACY_CARDINALITY_ESTIMATION	9481

Execution Plan Enhancements

- Shows query time statistics
  - CPU time and Elapsed time
  - SQL Server 2012 SP4, 2016 SP1, 2017, 2014 (TBD)
- Shows query-level Wait Stats!!
  - Top 10 wait stats
  - Also tracked in **sys.dm_exec_session_wait_stats**
  - SQL Server 2012 SP4, 2016 SP1, 2017 (compatibility 140!), 2014 (TBD)
- Shows query-Level Trace Flags
  - Including level: global, session or query
  - SQL Server 2012 SP4, 2014 SP2, 2016 SP1, 2017

Query execution profiling

sys.dm_exec_query_statistics_xml(session)

- Execution plan with transient, in-flight statistics
  - (e.g. row count, CPU)
- Available in SQL Server 2017 / SQL Server 2016 SP1
- Works both with standard and lightweight statistics profiling
  - E.g. execute **SET STATISTICS XML ON** before query to analyze

Query Store in SQL Server 2017

Wait Statistics integrated in Query Store!

sys.query_store_wait_stats

Organized in Wait Categories

ALTER DATABASE …
SET QUERY_STORE (WAIT_STATS_CAPTURE_MODE = ON)

Automatic Tuning

Automatic plan correction

- Automatically force last good plan when regression detected
- Minimum 10 CPU seconds improvement
- **sys.dm_db_tuning_recommendations** for manual corrections
- SQL Server 2017 and Azure SQL Database

Automatic index management

- Automatically **CREATE** and **DROP** indexes
- Verifies usage and performance improvements
- Azure SQL Database only (also automatic FORCE PLAN)

HA & DR

AlwaysOn FCI in newer releases

SQL Server 2016

- Group Managed Service Accounts (gMSA)
  - Managed directly by AD
  - Automatic password rotation

SQL Server 2017

- SQL Server on Linux support

AlwaysOn AG in SQL Server 2017

Cross database transactions supported (MSDTC)

Minimum number of (sync) replicas to commit

Clusterless support

CLUSTER_TYPE = NONE (e.g. read-only scale out replicas not used for HA)

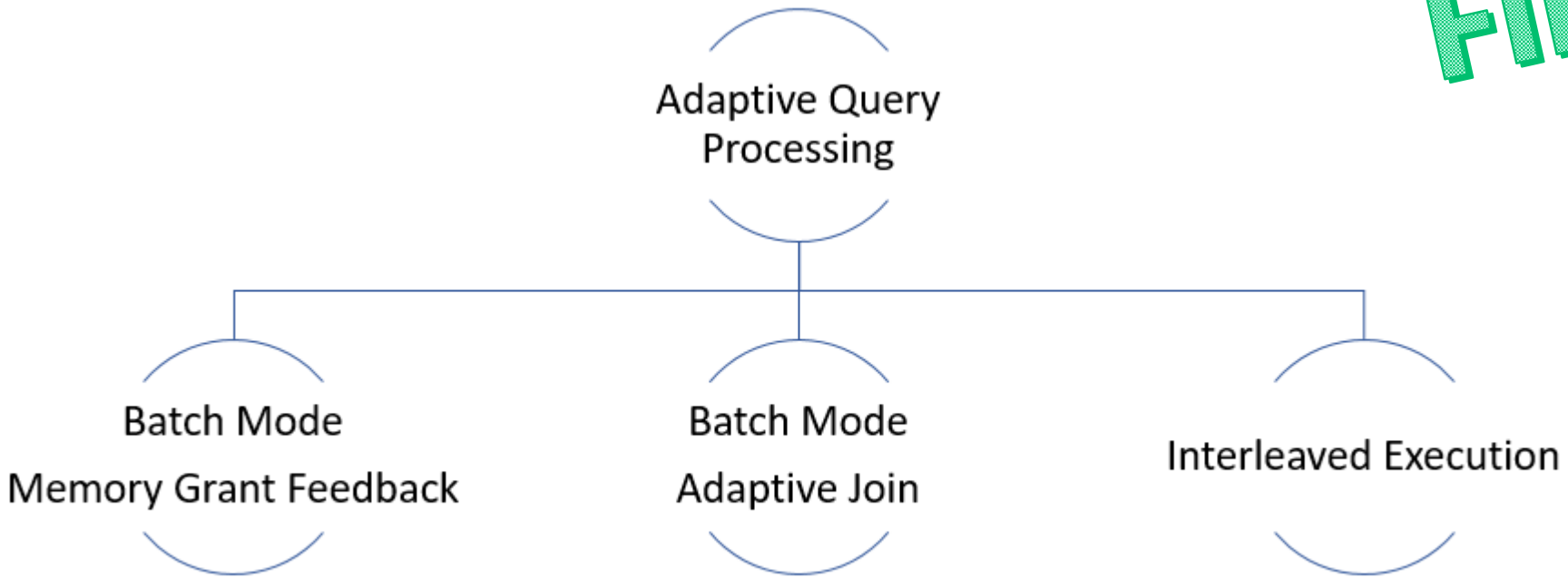
SQL Server on Linux support

CLUSTER_TYPE = EXTERNAL (Pacemaker)
CLUSTER_TYPE = NONE
Allows Windows-Linux cross-OS migrations

Performance

Adaptive Query Processing

Demo First!



Batch Memory Grant

Excessive Grant

Too much memory allocated vs. memory used
Impact: blocking, out-of-memory, reduced concurrency

Poor Grant

Not enough memory allocated resulting in data spill to tempdb
Impact: slow query, excessive disk usage (tempdb)

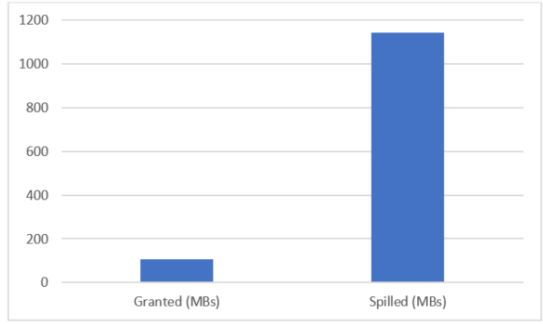
Grant increase

dynamic grants increase allocation too much
impact: server instability, unpredictable performance

Batch Mode Memory Grant Feedback

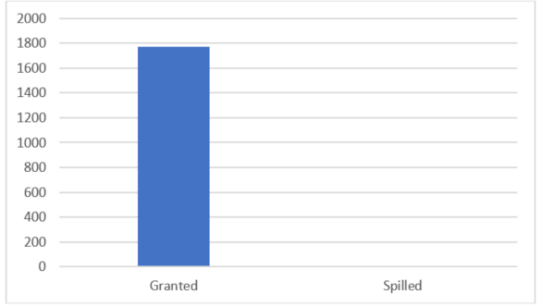
Post-execution evaluation

- Updates grant value for cached plan
- E.g. more memory if spilled, less if excessive grant



Parameter sensitive scenarios

- Some queries requires different plans with different grants
- Memory grant feedback will disable itself when unstable
  - **memory_grant_feedback_loop_disabled** extended event



Plan caching

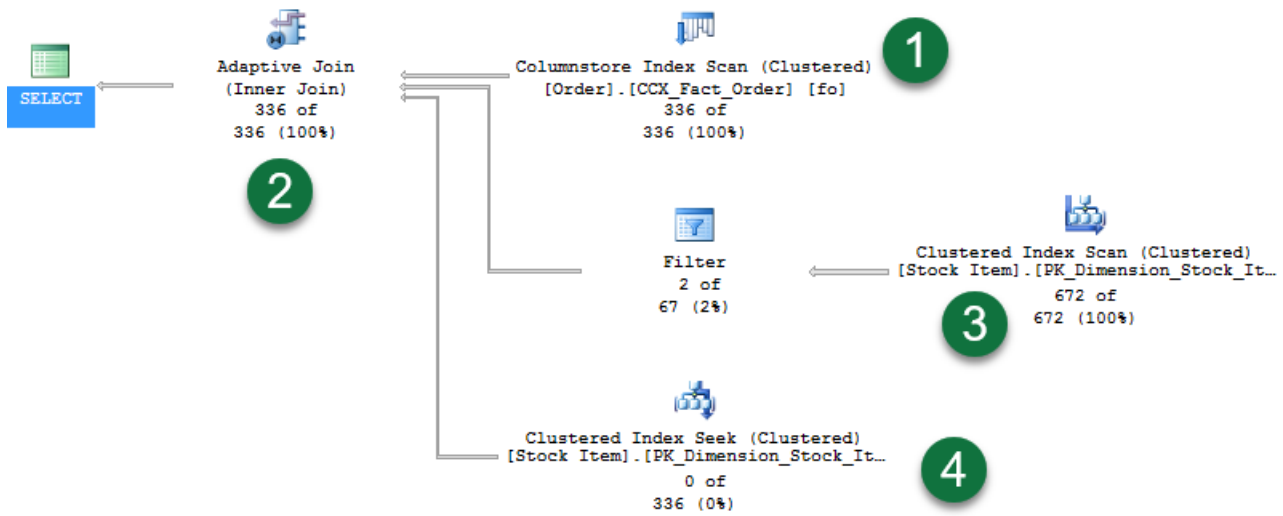
- Not persistent (i.e. not save in Query Store)
- OPTION(RECOMPILE) prevents caching and memory grant feedback

Batch mode adaptive joins

Scenario

- Nested loop algorithm better for small build join inputs
- Hash algorithm better for bigger inputs

Adaptive joins defer choice after first input scanned



Interleaved Execution

Problem with multi-statement table valued functions (MSTVFs)

- SQL Server <= 2012 optimize with cardinality = 1
- SQL Server 2014 & 2016 optimize with cardinality = 100

SQL Server >= 2017

- Start optimization
- Pause and executes MSTVFs if candidate
- Resume optimization with correct cardinality

Q&A