

Novità di SQL Server 2017

Chi sono

- 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)

Configurazione

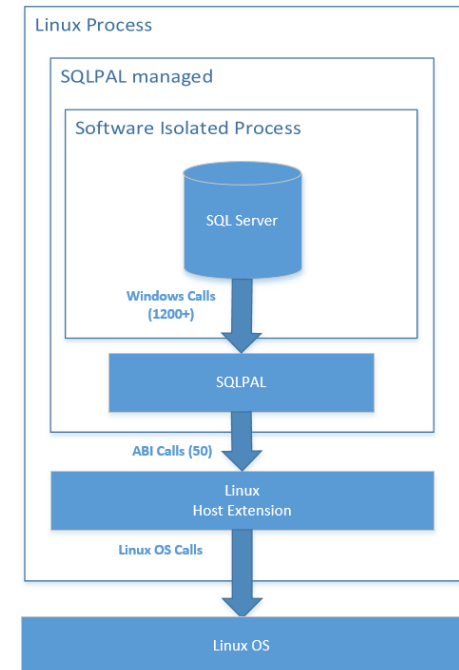
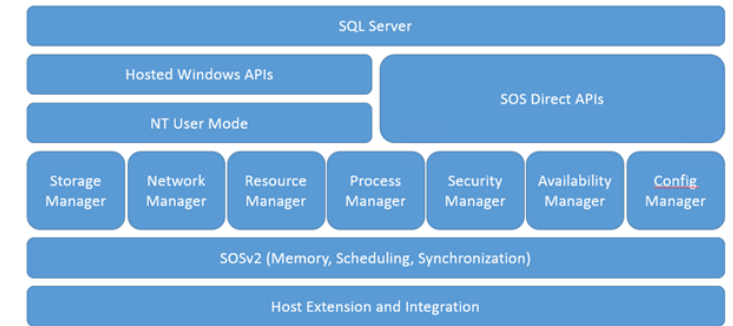
Novità di SQL Server 2017

Installazione

- Download separati
 - SQL Server Management Studio 17.x (usare anche per SQL Server 2008+)
 - SQL Server Data Tools
 - **Nuovo in 2017: Reporting Services**
- Configurazione **Tempdb**
 - Dimensioni file fino a 256GB
 - Avverte se > 1GB e «Instant File Initialization» non abilitata

Installazione SQL su Linux

- Facile
 - import repo GPG keys, register repository, apt-get/yum/zipper update/install
 - Docker
- **mssql-conf**
 - per creare script di installazione
 - usato da script di installazione
- Supporto a variabili di ambiente (mssql-conf/Docker)



Distribuzioni/versioni Linux supportate

- Red Hat Enterprise Linux 7.3 or 7.4
- Ubuntu Linux 16.0
- Suse Linux Enterprise Server v12 SP2
- Docker Engine 1.8+ on Linux, Mac, Windows

Requisiti minimi SQL su Linux

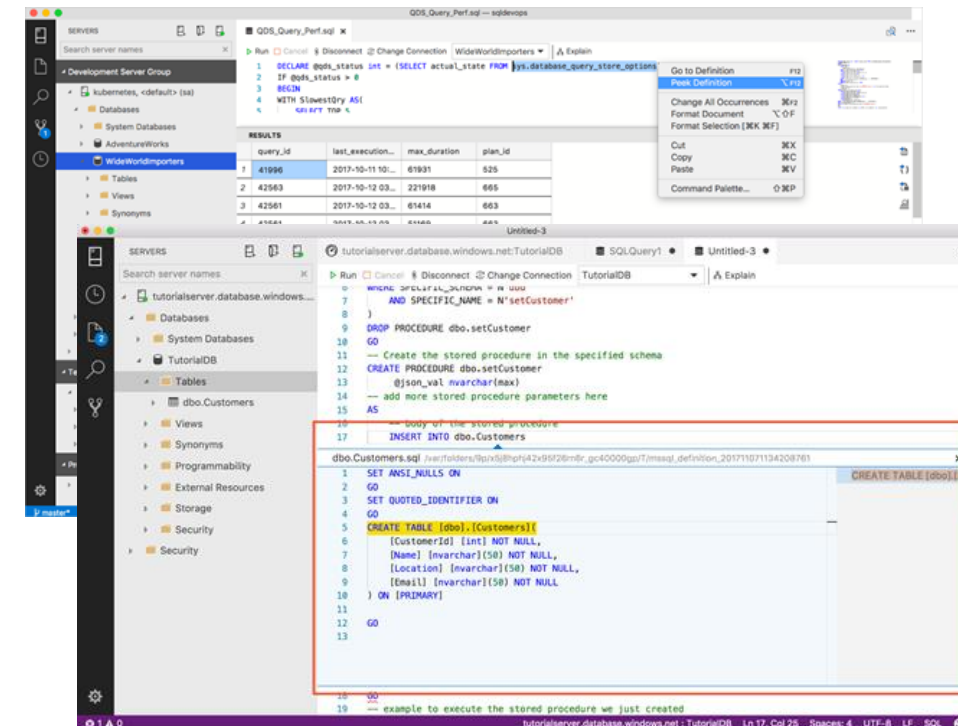
- 2 x core CPU 2Ghz (solo x64)
- 3,25 GB memoria
- 6GB spazio disco (file system XFS o EXT4)

Package SQL su Linux

- **mssql-tools**
 - librerie ODBC
 - **sqlcmd** per query da riga di comando
 - **bcp** per import/export «bulk»
- **mssql-server-agent**
- **mssql-server-fts**
 - Indicizzazione «Full-Text» e «Semantic Search» (richiede passi aggiuntivi)
- **mssql-server-is**

Strumentazione

- In Linux
 - Script configurazione con **mssql-conf**
 - Comandi T-SQL con **sqlcmd** o **mssql-cli**
- In Windows
 - SQL Server Management Studio (SSMS)
 - Visual Studio con SQL Server Data Tools (SSDT)
 - Visual Studio Code con estensione **mssql**
 - PowerShell
 - Comandi T-SQL con **sqlcmd**
- Multi-piattaforma (Windows, Mac, Linux)
 - SQL Operation Studio (Preview)



CLR Strict Security

- Code Access Security (CAS) in .NET Framework
 - Non più supportata come confine di sicurezza
 - Assembly con **PERMISSION_SET = SAFE** può accedere a risorse esterne, chiamare codice non gestito («unmanaged»), acquisire privilegi **sysadmin!**
- Nuova opzione di configurazione «CLR Strict Security»
 - Tratta **SAFE/EXTERNAL ACCESS** come **UNSAFE**
 - Richiede firma assembly e accesso con autorizzazione **UNSAFE ASSEMBLY**
 - (database con TRUSTWORTHY ON di proprietà di login con autorizzazione UNSAFE ASSEMBLY)
 - Abilitato di default può essere disattivato per compatibilità con versioni precedenti (non raccomandato)
- White-listing
 - **sp_add_trusted_assembly, sp_drop_trusted_assembly**

Configurazione Database

- **ALTER DATABASE SCOPED CONFIGURATION**

- Svuota la cache delle procedure
- Stima della cardinalità indipendente dal livello di compatibilità (TF 9481)
- «Parameter sniffing» (TF 4136)
- «Query optimization hotfixes» (TF 4199)
- **Nuovo in 2017:** cache IDENTITY, OFF per evitare «salti» (TF 272)

- **DATABASE SCOPED CREDENTIAL**

- Classe di protezione che supporta CONTROL, ALTER, REFERENCES, TAKE OWNERSHIP, VIEW DEFINITION

Sviluppo

Novità di SQL Server 2017

Database Development

Demo!

- **SELECT...INTO**
 - Supporta clausola ON per specificare filegroup destinazione
- Nuove opzioni per accesso BULK
 - **FORMAT = 'CSV'**
 - Supporto per RFC 4180 in BULK INSERT e OPENROWSET
 - **CREATE EXTERNAL DATA SOURCE ... TYPE = BLOB_STORAGE**
 - Punta a una specifica LOCATION URL
 - Usa «Shared Access Signatures» (SAS) con DATABASE SCOPED CREDENTIAL

Temporal Tables

- Supporto per **CASCADE DELETE** e **CASCADE UPDATE**
- Supporto criteri di conservazione
 - Anziché script eliminazione personalizzato, «Stretch Database» o partizionamento delle tabelle
 - Abilitazione a livello di database
 - ALTER DATABASE SET ... **TEMPORAL_HISTORY_RETENTION ON**
 - Specificato per tabella
 - Es. WITH(SYSTEM_VERSIONING = ON(... **HISTORY_RETENTION_PERIOD = 6 MONTHS**)
 - Supporto per DAYS, WEEKS, MONTHS, YEARS e INFINITE
 - Controllato contro colonna SYSTEM_TIME che rappresenta la fine del periodo

T-SQL Development

Demo!

- **CONCAT_WS**

- Concatena stringhe (colonne/valori) usando primo valore come separatore

- **TRANSLATE**

- Sostituisce insieme di caratteri in stringa con altro insieme di caratteri
- e.g. **SELECT TRANSLATE('2*[3+4]/{7-2}', '[]{}', '()()');**

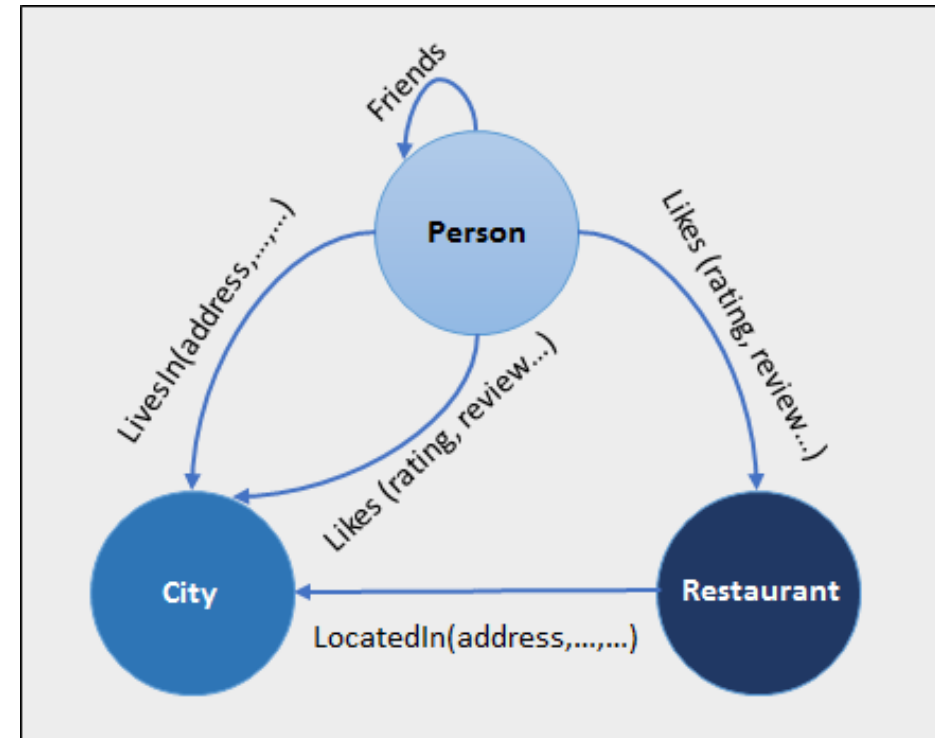
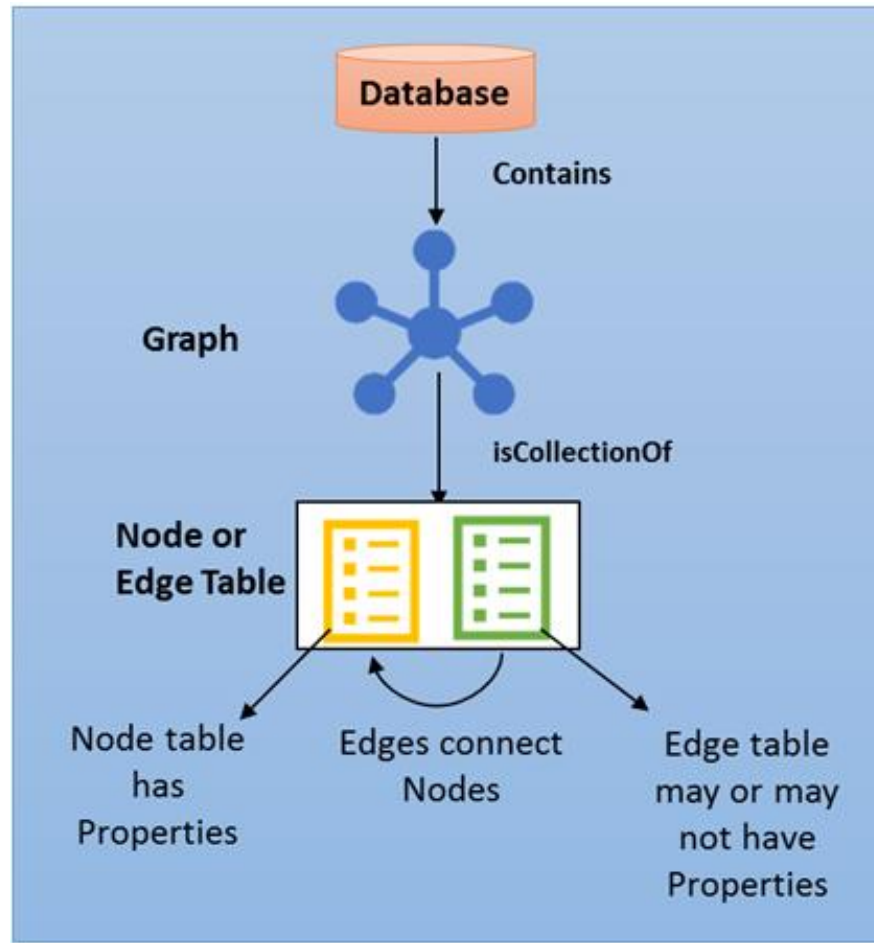
- **TRIM**

- Finalmente!!!! 😊
- Taglia spazi in testa/coda oppure un set di caratteri

- **STRING_AGG**

- Concatena stringhe (valori per riga) usando un separatore

«Graph Processing»



Concetti «Graph Processing»

- **Tabella «Node»**

- Rappresenta un entità

- **Tabella «Edge»**

- Rappresenta una relazione «many-to-many»
- Può avere proprietà
- Ha un orientamento (digrafo), connette due nodi

- **MATCH** comando T-SQL

- Condizione di ricerca per grafi
- «Pattern matching» e attraversamento

Limiti «Graph Processing»

- Tabelle temporanee locali/globali non supportate
- Tipi/variabili tabella non supportate
- Tabelle temporali «system-versioned» non supportate
- Tabella «memory optimized» non supportate
- Non si possono aggiornare direttamente (UPDATE) due «Node» collegati tramite un «Edge», soluzione
 - inserire nuove «Edge» che punta a differenti «Node»
 - cancellare precedente «Edge»
- Query «cross-database» con oggetti «Graph» non supportate

«In-Memory»

Novità di SQL Server 2017

Indici «Columnstore» in SQL Server 2012

- Solo un indice «non-Clustered Columnstore» per tabella
- Richiede sempre spazio extra (tabella rimane un B-Tree/Heap)
- Tabella non può esser aggiornata direttamente
- Molte limitazioni
 - Tipi dato
 - Schema
 - Interoperabilità con altre funzionalità

Indici «Columnstore» in SQL Server 2014

- «Clustered Columnstore»
 - Meccanismo di «storage» primario
 - Aggiornabile
 - Nessun altro indice consentito su tabella
- «Non-Clustered Columnstore»
 - Come in SQL Server 2012
- Opzione di compressione «Archival»
- Più operatori in modalità «Batch»
 - Scan, Filter, Project, Join, Group By, Union All
 - Ancora solo query parallele

Indici «Columnstore» in SQL Server 2016

- «Clustered Columnstore»
 - Uno o più indici «non-Clustered Rowstore» supportati
 - PK e FK possono essere supportati da indici «non-Clustered Rowstore»
- «Non-Clustered Columnstore»
 - Supporto per un solo indici aggiornabile su tabelle «Rowstore»
 - Supporto per condizione di filtro (es. dati «caldi» di un carico di lavoro OLTP)
- Tabelle «In-Memory» possono avere un indice «Columnstore»
- «Compression Delay»
 - Evita frammentazione quando dati caldi cambiano frequentemente con carichi di lavoro OLTP
- Ancora più operatori in modalità «Batch»
 - sort, aggregates with multiple distinct functions, window aggregate functions, window user-defined aggregates, window aggregate analytic functions
 - Supporto per query «Single-Threaded» (MAXDOP 1)

Indici «Columnstore» in SQL Server 2016

- Supporto per livelli di isolamento transazionale RCSI e SI
- Deframmentazione indici tramite riorganizzazione
- Supporto per repliche secondarie AlwaysOn AG
- «Pushdown» aggregazioni e predicati su stringhe (durante scansioni)
- Disponibile in tutte le edizioni di SQL Server 2016 Service Pack 1

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	No**

SQL Server 2016 SP1 Edition	Columnstore Index Quota (per Instance)
Express	352 MB
Web	16 GB
Standard	32 GB
Developer	Unlimited
Enterprise	Unlimited

Indici «Columnstore» in SQL 2017

- Supporto per colonne calcolate non persistenti
- Supporto per LOB
 - Solo «Clustered Columnstore»
 - Già disponibile anche in Azure SQL Database (Premium)
 - Es. caso di studio utilizzo dati JSON
 - <https://blogs.msdn.microsoft.com/sqlserverstorageengine/2017/02/09/json-data-in-clustered-column-store-indexes>
- Supporto «online build/rebuild» indici «non-clustered Columnstore»
- Riepilogo funzionalità per versione prodotto
 - <https://docs.microsoft.com/en-us/sql/relational-databases/indexes/columnstore-indexes-what-s-new>

Tabelle «Memory-Optimized» in SQL 2016

- Formato «Storage» cambiato
- Molti miglioramenti
 - Supporto per aggiornamento e campionamento statistiche
 - ALTER TABLE eseguito in parallelo e scrittura solo metadati nel T-Log
 - Miglioramenti T-SQL
 - Es. UNIQUE, FK, CHECK, TRIGGERS (solo AFTER), LOB (colonne)
 - Esteso supporto moduli T-SQL compilati nativamente
 - Es. UNION, UNION ALL, DISTINCT, OUTER JOIN, «Subqueries», OUTPUT, LOB (variabili/parametri), «Inline-TVF» e «Inline-UDF» compilate
 - Supporto per piani di esecuzione paralleli
 - <https://docs.microsoft.com/en-us/sql/database-engine/whats-new-in-sql-server-2016#InMemory>

Tabelle «Memory-Optimized» in SQL 2017

- Supporto colonne calcolate
 - Incluso indici su colonne calcolate
- Supporto funzioni JSON
 - Sia moduli T-SQL compilati nativamente che vincoli CHECK
- Esteso supporto moduli T-SQL compilati nativamente
 - CROSS APPLY, CASE, TOP (N) WITH TIES, sp_spaceused, sp_rename
- Prestazioni
 - Fase REDO ripristino log delle transazioni effettuata in parallelo

Amministrazione

Novità di SQL Server 2017

Metadati di Sistema

Demo!

- **sys.dm_os_sys_info**
 - Aggiunti **socket_count**, **cores_per_socket**, **numa_node_count**
 - Utile con VM
- **sys.dm_os_host_info**
 - Informazioni SO su Windows/Linux
- Non documentate
 - **sys.dm_os_enumerate_fixed_drives**
 - **sys.dm_os_enumerate_filesystem(dir, pattern)**
 - **sys.dm_os_file_exists(dir_or_file)**

Metadati Database

Demo!

- **sys.dm_db_log_stats()**
 - Informazioni di riepilogo transaction-log, utile per monitoring
- **sys.dm_db_file_space_usage**
 - Aggiunta colonna **modified_extent_page_count**
 - Soluzioni di «smart backup» es. DIFF se modifiche < 70-80%, FULL altrimenti
- **sys.dm_tran_version_store_space_usage**
 - Spazio riservato in pagine/KB
- **sys.dm_db_stats_histogram(object, stat)**
 - SQL Server 2017 / SQL Server 2016 SP1 CU2

DBCC CLONEDATABASE

- Crea una copia vuota del database per «troubleshooting»
 - Niente dati ma schema completo (vincoli, tipi, moduli T-SQL, ecc.)
 - Statistiche e Query Store!
 - Non Bloccante
 - In sola lettura come impostazione predefinita, ma può essere modificato
 - Opzionale **NO_STATISTICS, NO_QUERYSTORE**
 - Disponibile in SQL Server 2012 SP4, 2014 SP2 CU3, 2016 SP1, 2017

Opzione query «USE HINT»

- «Hint» denominati al posto di Trace Flag difficili da ricordare
 - **OPTION (USE HINT('DISABLE_PARAMETER_SNIFFING'))** al posto di **OPTION (QUERYTRACEON 4136)**
 - Non richiede permessi **sysadmin!**
 - Disponibile in SQL Server 2016 SP1, 2017
- Lista disponibile con **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

Miglioramenti piani di esecuzione

Demo!

- Tempi di esecuzione query
 - «CPU time» e «Elapsed time»
 - SQL Server 2012 SP4, 2016 SP1, 2017, 2014 (TBD?)
- Top 10 «Wait Stats» a livello di query
 - Attenzione all'accuratezza
 - <https://www.brentozar.com/archive/2017/07/sql-2016-sp1-shows-wait-stats-execution-plans>
 - https://blogs.msdn.microsoft.com/sql_server_team/making-parallelism-waits-actionable
 - SQL Server 2016 SP1, 2017 (compatibilità 140!)
- «Trace Flags» attive
 - Include il livello: global, session o query
 - SQL Server 2012 SP4, 2016 SP1, 2017

Profilazione esecuzione query

Demo!

- **sys.dm_exec_query_statistics_xml(session)**
 - Piano di esecuzione con statistiche transitorie «in-flight» (es. n° righe, CPU)
 - Funziona con profilazione
 - Standard (eseguire SET STATISTICS XML ON prima di eseguire la query)
 - «lightweight statistics» (attivare Trace Flag 7412)
 - Disponibile in SQL Server 2016 SP1, 2017

Query Store in SQL Server 2017

Demo!

- «Wait Stats» integrate con Query Store!
 - Nuova DMV **sys.query_store_wait_stats**
 - Organizzate in categorie
 - **ALTER DATABASE ...
SET QUERY_STORE (WAIT_STATS_CAPTURE_MODE = ON)**

«Tuning» automatico

- Correzione dei piani automatica
 - Forza ultimo piano ottimale quando identifica regressione
 - Deve esserci miglioramento minimo di 10 secondi «CPU»
 - Usare **sys.dm_db_tuning_recommendations** per correzioni manuali
 - Disponibile in SQL Server 2017 e Azure SQL Database
- Gestione automatica indici
 - Creazione/eliminazione indici
 - Verifica effettivo utilizzo e miglioramento (modello ML)
 - Disponibile solo in Azure SQL Database

HA & DR

Novità di SQL Server 2017

AlwaysOn FCI nelle release recenti

- **SQL Server 2016**

- «Group Managed Service Accounts» (gMSA)
 - Gruppi gestiti direttamente da AD
 - Rotazione delle password automatica

- **SQL Server 2017**

- Supporto per Linux

AlwaysOn AG in SQL Server 2014

- Fino a 8 repliche secondarie (era 4)
 - Comunque solo 2 in modalità sincrona
- Repliche secondarie rimangono in linea
 - quando disconnesse dalla primaria
 - quando c'è perdita di Quorum
- Nuova DMV
 - `sys.fn_hadr_is_primary_replica`
- Nuovo Wizard «Add Azure Replica»

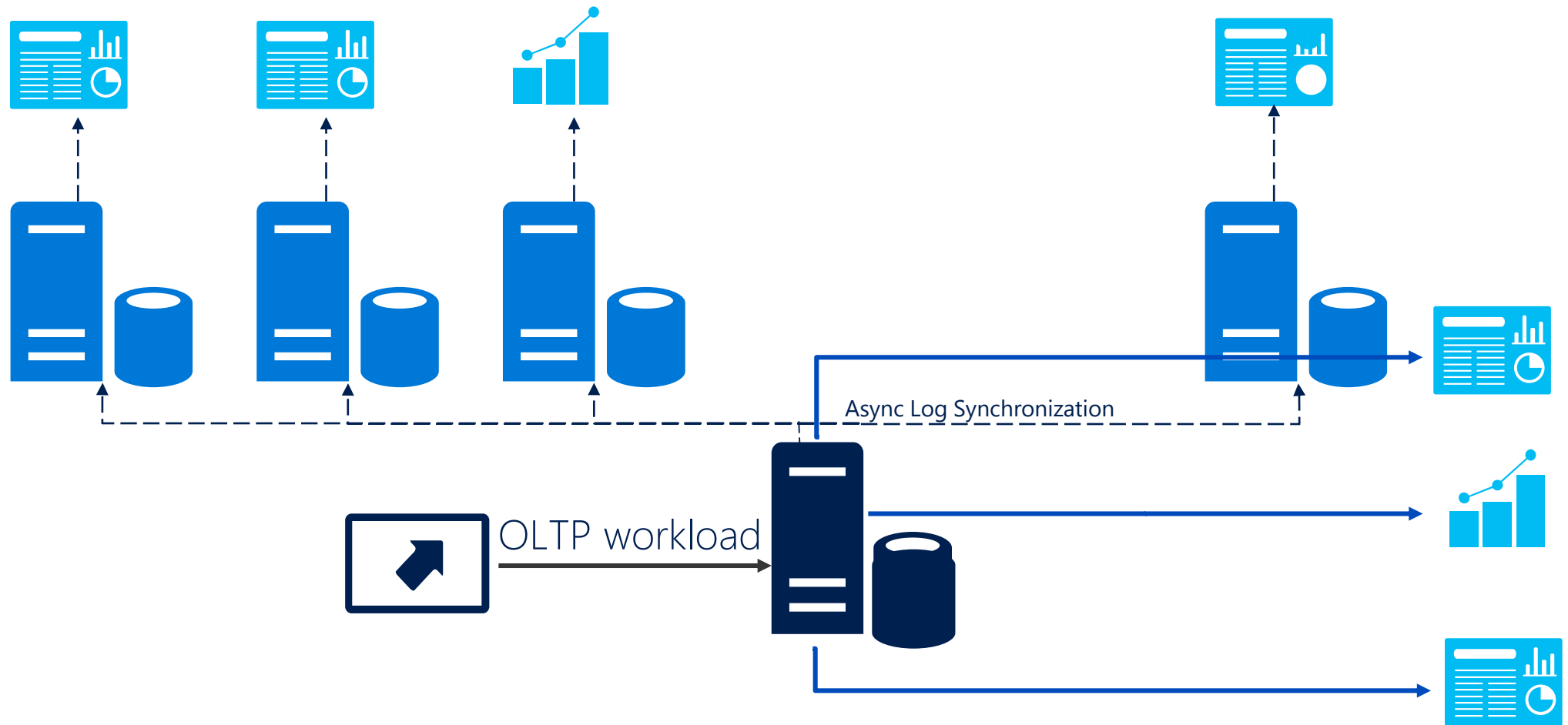
AlwaysOn AG in SQL Server 2016

- Fino a 3 repliche sincrone con «Failover» automatico
- «Failover» database migliorato
 - fallimento a fronte di errori scrittura transazione
- «Seeding» replica secondaria (al posto di Backup/Restore)
- Repliche secondarie «Read-Only» per «Load Balancing»
- MSDTC supportato con Windows Server 2016 (solo «Cross-Istance»)
- Supporto database crittografati (es. SSIDB)
- «Cross-Domain» e «Domain-Independent» AG
- «Distributed» AG
- Trasporto dei «Log Record» riscritto per essere molto più veloce

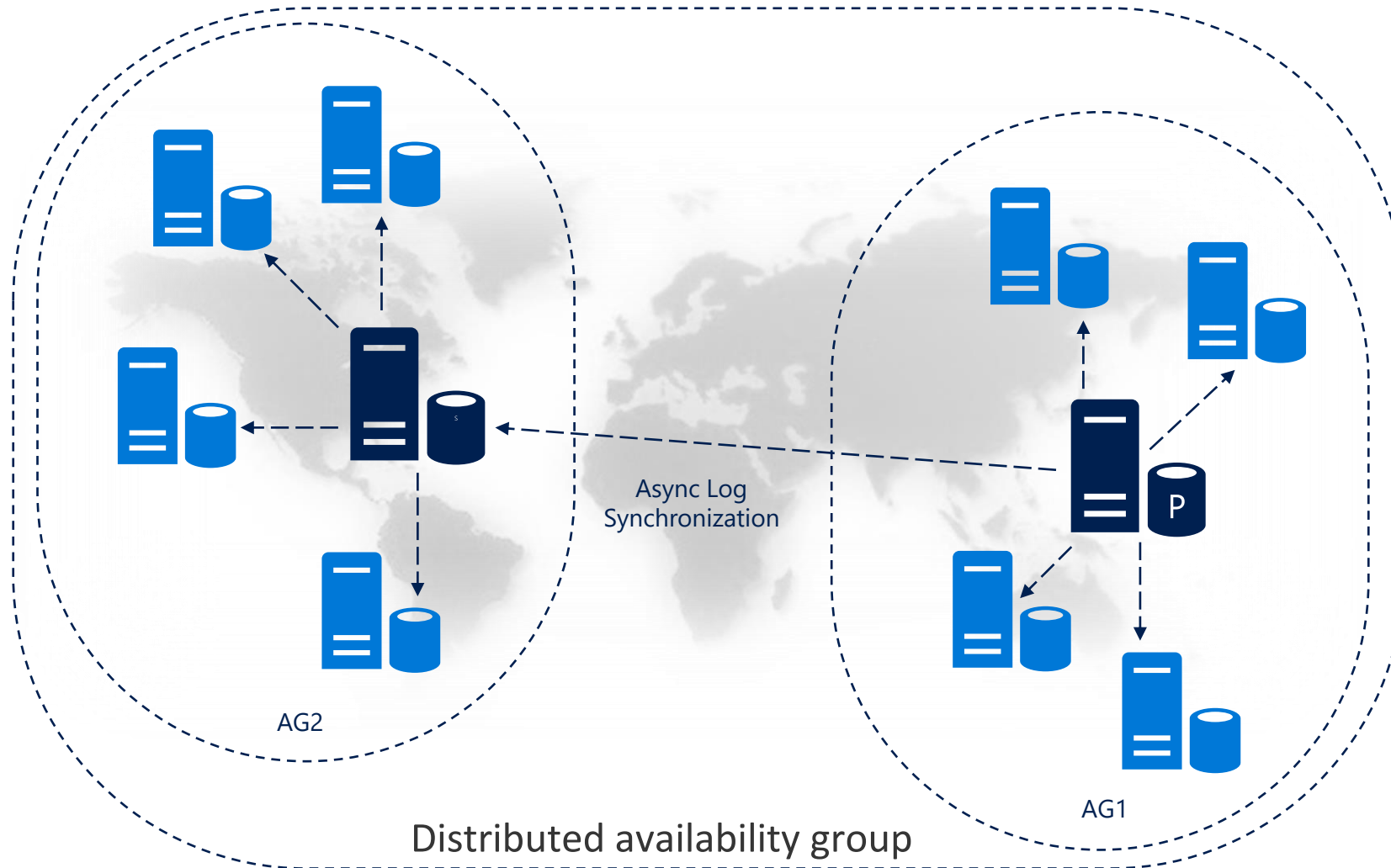
AlwaysOn Basic AG in SQL Server 2016

- Disponibile con Edizione Standard
 - Soluzione per Database Mirroring deprecato (quasi...)
- Limite a 2 repliche
 - Sincrone o asincrone (DBM richiede EE per asincrone!)
 - Replica può essere in Azure
- Limite di 1 database per Availability Group...
- Replica secondaria non accessibile
 - es. «Read-Only», Backup...

Scalabilità con repliche «Read-Only» in SQL 2016



Scalabilità con «Distributed» AG in SQL 2016



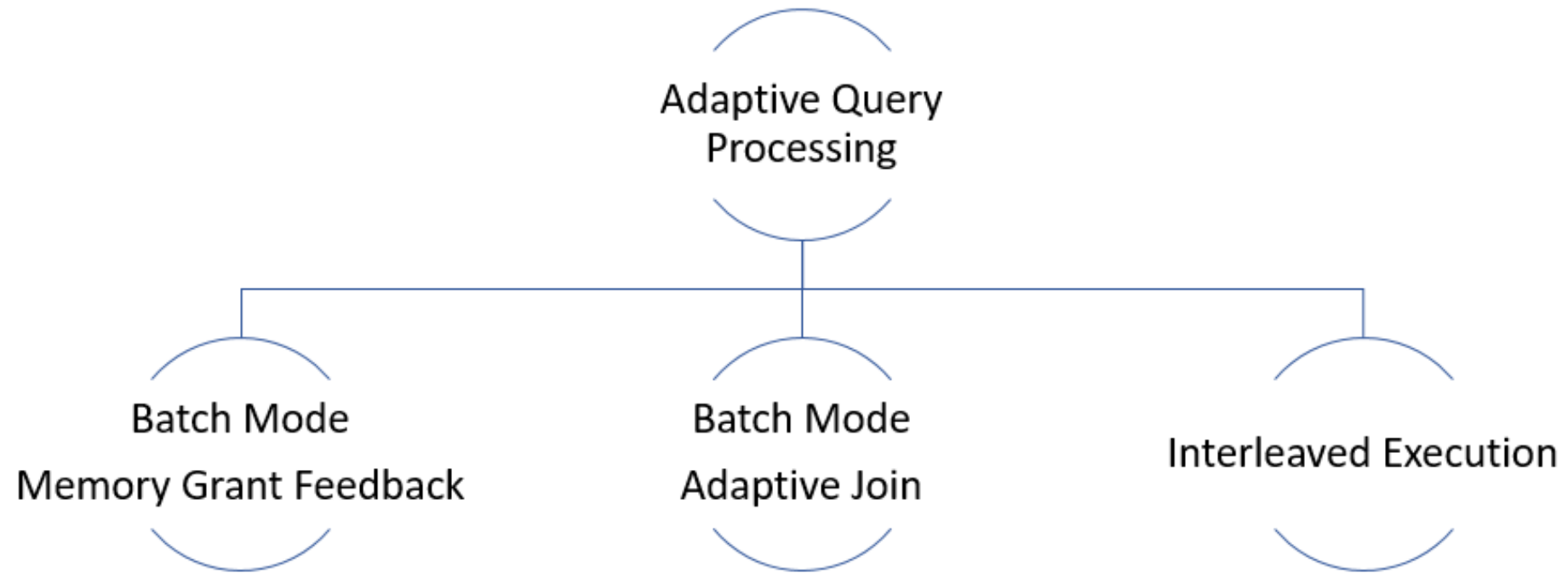
AlwaysOn AG in SQL Server 2017

- Supporto transazioni «cross-database» (MSDTC)
- Numero minimo di repliche (sincrone) che devono confermare
- Supporto scenari «Clusterless»
 - **CLUSTER_TYPE = NONE** (e.g. read-only scale out replicas not used for HA)
- Supporto per Linux
 - **CLUSTER_TYPE = EXTERNAL** (Pacemaker)
 - **CLUSTER_TYPE = NONE**
 - Permette anche migrazioni cross-SO Windows-Linux

Prestazioni

Novità di SQL Server 2017

«Adaptive Query Processing»

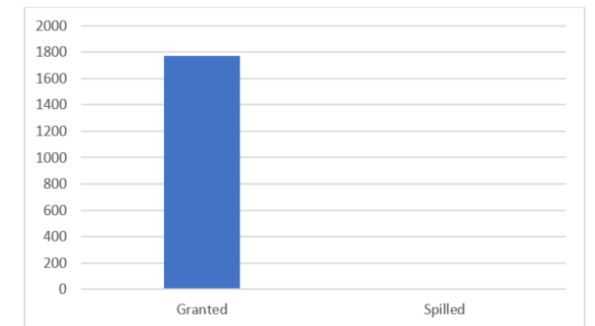
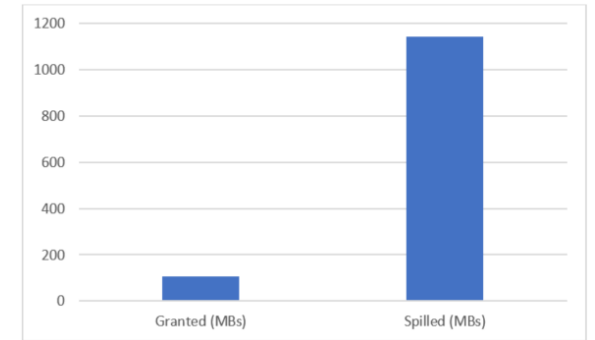


«Batch Memory Grant»

- «Excessive Grant»
 - Troppa memoria allocata rispetto a quanto utilizzata
 - Impatto: «blocking», «out-of-memory», concorrenza ridotta
- «Poor Grant»
 - Memoria allocata insufficiente con conseguente «spill» nel tempdb
 - Impatto: query lenta, eccessivo utilizzo del disco (tempdb)
- «Grant increase»
 - Incremento memoria allocata dinamicamente troppo grande
 - Impatto: instabilità del server, performance non predicibili

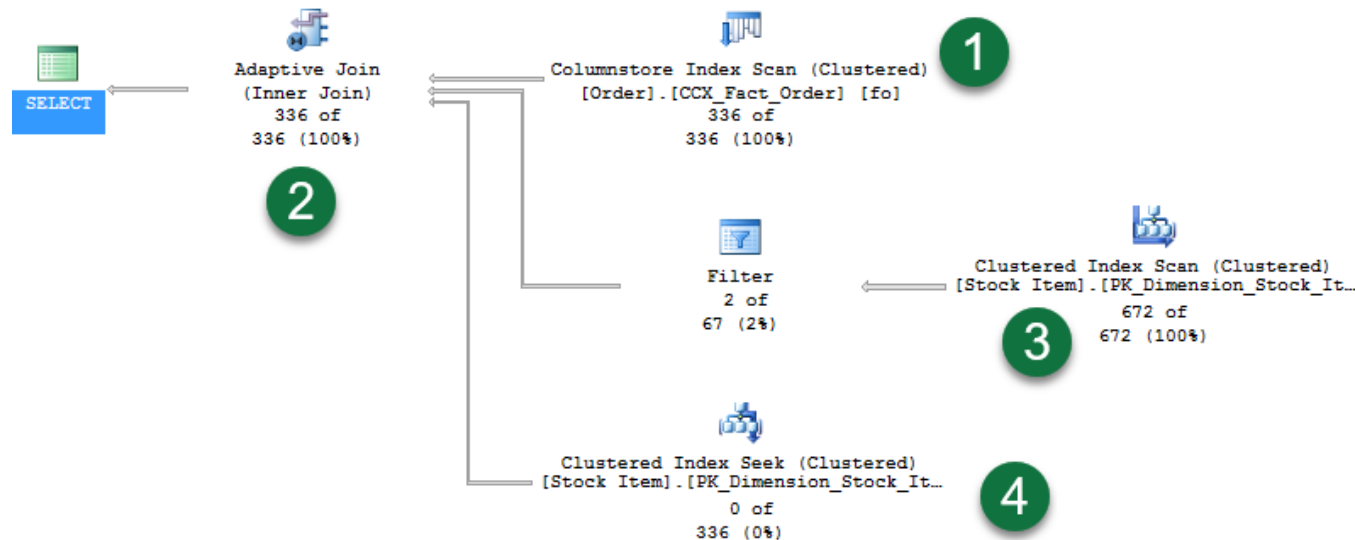
Batch Mode Memory Grant Feedback

- Valutazione post-esecuzione
 - Aggiorna il valore di allocazione dei piani in cache
 - Es. più memoria se «Poor», meno se «Excessive»
- Scenario sensibili a valori parametri
 - Alcune query piani con allocazione diversa
 - «memory grant feedback» si disabilita quando non stabile
 - **memory_grant_feedback_loop_disabled** extended event
- Memorizzazione nella cache dei piani
 - Non persistente (non salvata nel Query Store)
 - OPTION(RECOMPILE) impedisce memorizzazione e «memory grant feedback»



«Batch mode adaptive joins»

- Scenari join
 - Algoritmo «nested loop» migliore con piccole tabelle candidate come «build»
 - Algoritmo «hash» migliore con grandi tabelle candidate come «build»
- «Adaptive joins» rinvia la scelta dopo scansione primo input



«Interleaved Execution»

- Problema con «multi-statement table valued functions» (MSTVF)
 - SQL Server <= 2012 ottimizza con cardinalità fissa = 1
 - SQL Server 2014 & 2016 ottimizza con cardinalità fissa = 100
- SQL Server >= 2017
 - Ottimizzazione parte
 - Ottimizzazione viene in pausa per eseguire MSTVF (se candidata)
 - Ottimizzazione riprende con cardinalità corretta

Domande & Risposte