

SQL Server

Query tuning in pillole

Danilo Dominici

ddominici@gmail.com

@danilo_dominici



#566 | PARMA 2016

Sponsors



November 26°, 2016

Organizers



UNIVERSITÀ DEGLI STUDI DI PARMA



getlatestversion.it



dotNET {podcast}

ENGAGE
IT SERVICES



dan|ela
ma|visi
COMMUNICATION



#sqlsatParma
#sqlsat566

November 26°, 2016

Chi sono

- Consulente, Trainer, Speaker, Autore
- Uso SQL Server da... un sacco di tempo!
- Microsoft MVP e MCT
- Mentor @ SolidQ
- DBA SQL Server e VMWare admin
@ Regione Marche
- Co-leader del PASS Global Italian Virtual Chapter
<http://globalitalian.sqlpass.org>



Microsoft
CERTIFIED
Trainer



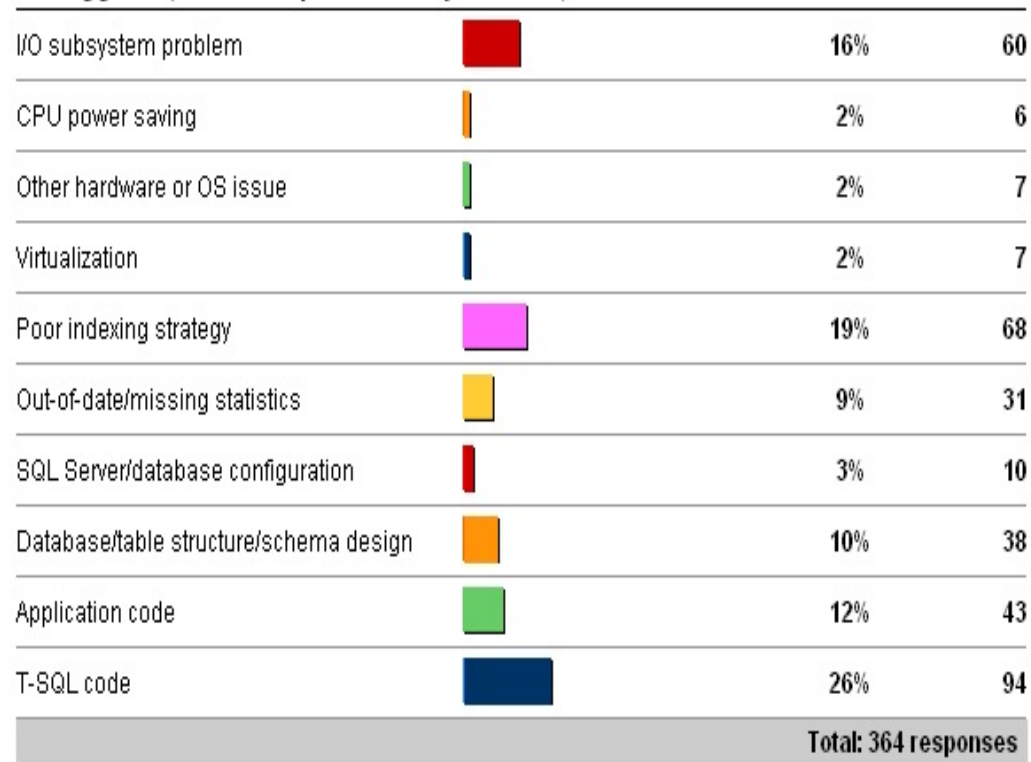
November 26°, 2016

Query tuning: perché?

- Segnalazioni da parte degli utenti
 - C'è sempre almeno un report che il capo vuole per ieri 😊
- Performance variabili nel tempo
 - incremento di volume dei dati
 - modifiche alle strutture del db
 - scelte di sviluppo errate
 - utilizzo di nuove funzionalità

Query tuning: perché?

What were the root causes of the last few SQL Server performance problems you debugged? (Vote multiple times if you want!)



<http://sqlskills.com/blogs/paul/post/survey-results-common-causes-of-performance-problems.aspx>

La cassetta degli attrezzi

- SQL Server DMVs
- SQL Trace / Extended Events
- Plan Explorer di SQL Sentry
- sp_whoisactive di Adam Machanic
- sp_Blitzcache di Brent Ozar
- Diagnostic Queries di Glenn Berry



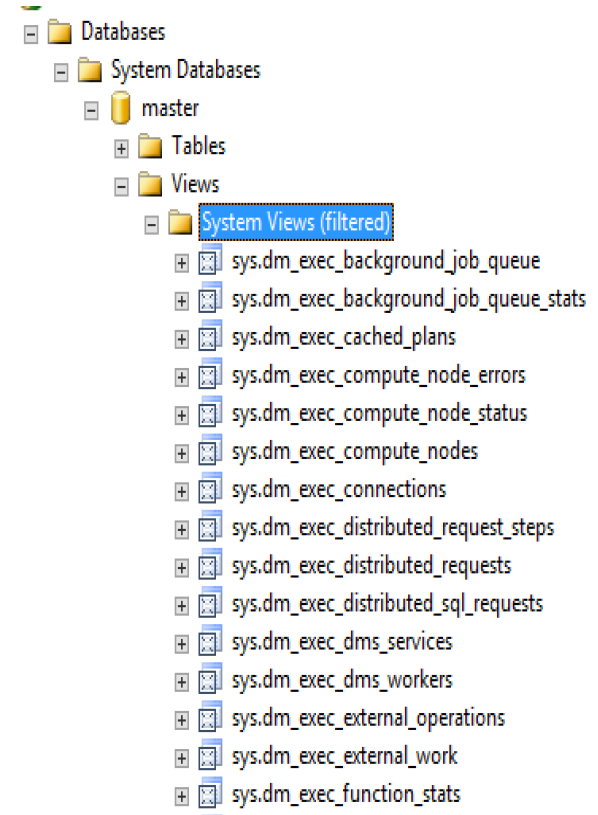
November 26°, 2016



SQL Server DMVs



- Viste o funzioni che accedono ai metadati di SQL Server
- Categorizzate per funzione
 - `sys.dm_exec_*`
 - `sys.dm_index_*`
- Vanno generalmente combinate per avere risultati esaustivi



SQL Trace



- SQL Trace
 - Consente di registrare il workload di SQL Server per un periodo di tempo su file (*) o database
 - Attenzione a NON utilizzare lo stesso disco dove sono i dati per registrare la traccia !!!
 - In combinazione con la funzionalità **Distributed Replay** di SQL Server, consente di effettuare il replay di un workload - <http://bit.ly/2grAuk2>
 - Ad esempio per valutare l'impatto delle modifiche effettuate sul database o sull'applicazione
 - O per valutare la portabilità di un database su versioni successive di SQL Server
 - Deprecato, a partire da SQL Server 2012, in favore degli Extended Events – ma ancora utilizzabile

SQL Profiler



- SQL Profiler
 - E' la GUI di SQL Trace
 - NON lanciarlo sul server
 - Consuma memoria e spazio disco

Extended events

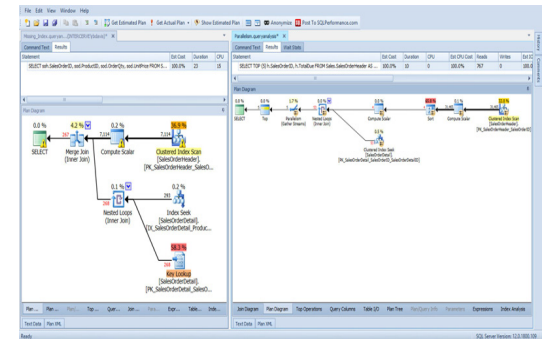


- Sistema di monitoring leggero e a basso costo
- Flessibile e configurabile
- Può essere lasciato in esecuzione senza impattare sulle performance
 - Esiste anche una sessione di SQL Server denominata System Health
- Utilizzabile sia da GUI che via T-SQL
- Posso scrivere su diverse destinazioni
 - Anche in parallelo

SQL Sentry Plan Explorer



- Tool per l'analisi dei piani di esecuzione delle query e degli indici di SQL Server
 - Rilasciata da poche settimane la versione unica che riunisce la versione gratuita e la versione PRO
- Download <http://bit.ly/2cMJmyo>



sp_whoisactive



- Stored procedure che attinge da diverse DMVs
 - Scritta da Adam Machanic – Data Platform MVP
 - Download <http://bit.ly/2dorOuU>
- Fornisce moltissime informazioni sulle query in esecuzione
 - Testo della query, wait info, numero di letture e scritture, memoria impegnata, blocchi, piano di esecuzione, etc

| | session_id | sql_text | wait_info | blocking_session_id | blocked_session_count |
|---|------------|---|-------------------|---------------------|-----------------------|
| 1 | 53 | <?query -- UPDATE TOP(10) Sales.SalesOrderDet... | NULL | NULL | 3 |
| 2 | 54 | <?query -- UPDATE TOP(10) Sales.SalesOrderData... | (721028ms)LCK_M_U | 53 | 2 |
| 3 | 55 | <?query -- UPDATE TOP(10) Sales.SalesOrderData... | (718711ms)LCK_M_U | 54 | 0 |
| 4 | 56 | <?query -- UPDATE TOP(10) Sales.SalesOrderData... | (716748ms)LCK_M_U | 54 | 0 |

sp_blitzcache



- Analizza la cache ed evidenzia le query problematiche

Results

Messages

| | Database | Cost | Query Text | Query Type | Warnings | # Executions | Executions / Minute | Execution V |
|---|--------------------|-----------|---|------------|--|--------------|---------------------|-------------|
| 1 | AdventureWorks2014 | 11.169 | SELECT p.Name AS ProductName, NonDiscountSales ... | Statement | | 1 | 0.00 | 0.3195 |
| 2 | AdventureWorks2014 | 0 | SELECT 'Total income is', ((OrderQty * UnitPrice) * (1.0 - U... | Statement | | 1 | 0.00 | 0.3195 |
| 3 | AdventureWorks2014 | 2.5587 | SELECT udf.name AS [Name], udf.object_id AS [ID], udf.... | Statement | Compilation Timeout, Plan Warnings, Implicit Co... | 1 | 0.00 | 0.3195 |
| 4 | WideWorldImporters | NULL | select p.object_id as object_id from ((select o.object_id a... | Statement | Unused Memory Grant | 140 | 2.00 | 44.7284 |
| 5 | AdventureWorks2014 | NULL | with QueryAggregation as (select top(100) query_hash, ... | Statement | | 1 | 0.00 | 0.3195 |
| 6 | WideWorldImporters | NULL | DELETE TOP (1000000) FROM [sys].[memory_optimized_... | Statement | | 140 | 2.00 | 44.7284 |
| 7 | AdventureWorks2014 | 0.0311451 | SELECT * FROM Production.Product ORDER BY Name... | Statement | | 1 | 0.00 | 0.3195 |
| 8 | AdventureWorks2014 | 0.0872922 | SELECT clms.name AS [Name], clms.column_id AS [ID]... | Statement | Plan Warnings | 23 | 0.00 | 7.3482 |

| | Priority | FindingsGroup | Finding | URL | Details | CheckID |
|---|----------|----------------------|------------------------------------|--|---|---------|
| 1 | 50 | Execution Plans | Compilation timeout | http://brentozar.com/blitzcache/compilation-timeo... | Query compilation timed out for one or more querie... | 18 |
| 2 | 50 | Execution Plans | Query Plan Warnings | http://brentozar.com/blitzcache/query-plan-wami... | Warnings detected in execution plans. SQL Serve... | 8 |
| 3 | 50 | Performance | Implicit Conversions | http://brentozar.com/go/implicit | One or more queries are comparing two fields that ... | 14 |
| 4 | 100 | Unused memory grants | Queries are asking for more mem... | No URL yet. | Queries have large unused memory grants. This c... | 30 |

Glenn Berry's Diagnostic Queries



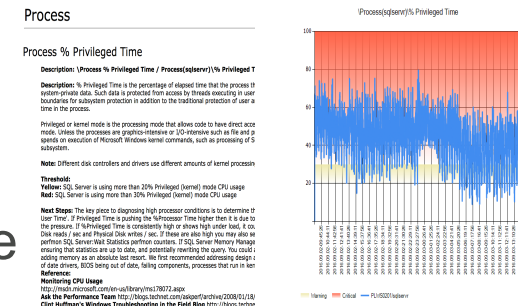
- Raccolta di query utili per lo “screening” di SQL Server
 - Aggiornate mensilmente (bug, nuove implementazioni, aggiornamento delle info su CU/ Service Packs)
 - Download <http://bit.ly/Q5GAJU>
 - Info dettagliate sulle principali query utilizzate in una serie di post: <http://bit.ly/2cMJtKo>

Da dove iniziare ?

- Prima di tutto fate una verifica del server
 - Configurazione server (# processori, NUMA, controller e layout dischi, formattazione volumi SQL Server, power plan di Windows)
 - Configurazione SQL Server (max memory, parallelismo, tempdb, ad-hoc queries, Instant file initialization, filegrowth)
- Common non-configured options on a Database Server di Sergio Govoni
 - <http://bit.ly/2gJPVbq>

Baseline

- Registrare una baseline
 - Possibilmente in tempi non “sospetti” 😊
 - Se già ci sono problemi di performance sarà comunque utile per verificare i miglioramenti introdotti
- Tools
 - Windows Performance Monitor
 - SQL Trace
 - Performance Analysis of Logs (PAL)
 - Usa un sacco di CPU...
 - Opzione 2: importare i dati con relog.exe PowerPivot in Excel e farci le analisi



Identificare le query lente

- Identificare quelle che hanno valori elevati di logical reads, CPU, durata, costo della query
 - dalla baseline
 - DMVs
 - Extended events / SQL Trace
 - sp_blitzcache
 - sp_whoisactive
 - Su segnalazione degli utenti ☺

Preparare l'ambiente di test

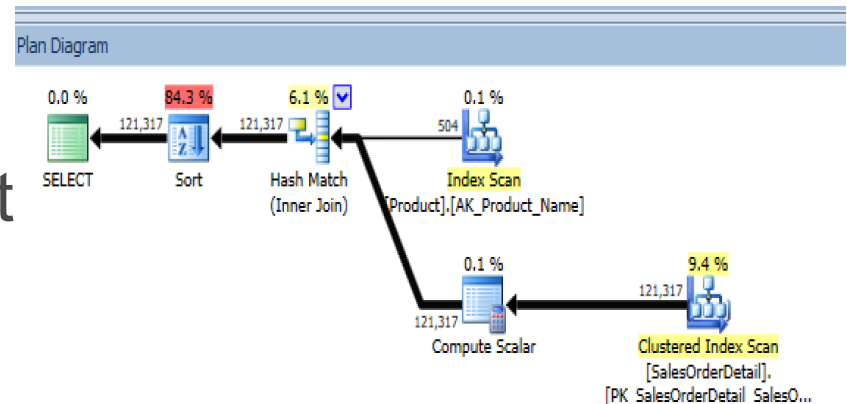
- **DBCC FREEPROCCACHE**
 - Svuota la cache delle stored procedure
- **DBCC FREESYSTEMCACHE**
 - Svuota l'intera cache di SQL Server
- **DBCC DROPCLEANBUFFERS**
 - Svuota la cache dei dati
- **SET STATISTICS IO**
 - Abilita la visualizzazione delle metriche di I/O della query
- **SET STATISTICS TIME**
 - Abilita la visualizzazione dei tempi di esecuzione della query

Metodologia spicciola

- Eseguire la query da SSMS con gli switch per il tuning on
 - SET STATISTICS IO ON
 - SET STATISTICS TIME ON
 - Include actual execution plan
- Salvare il piano di esecuzione originale
- Applicare le variazioni e confrontare i piani risultanti con quello originale
 - Nuova funzionalità Compare Plan del SSMS di SQL Server 2016

Analisi del piano di esecuzione

- Che cosa cercare:
 - Scans
 - Missing indexes
 - Key/RID lookup
 - Sort
 - Distinct
 - Righe stimate vs. effet
 - Conversioni implicite
 - Parallelismo
 - Spills / spool

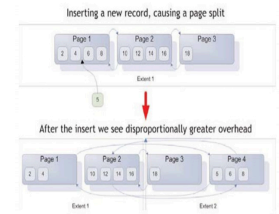


I problemi più comuni

- Indicizzazione non corretta
- Statistiche non accurate
- Parameter sniffing
- Cursori

Indicizzazione non corretta

- Indici mancanti
 - Scan dell'intera tabella – maggiore I/O e RAM
 - Parallelismo
- Indici frammentati
 - Per monitorare la frammentazione:
 - DBCC SHOWCONTIG(<tablename>) – Legacy ☺
 - sys.dm_db_index_physical_stats
 - Per ridurre o eliminare la frammentazione
 - Maintenance Plan – Reorganize/Rebuild task - ☹
 - Custom solutions ☺
 - Ola Hallengren - <https://ola.hallengren.com>
 - Minion Reindex - <http://minionware.net/reindex/>
 - AdaptiveIndexDefrag - <http://bit.ly/2flQl1g>



Statistiche non accurate

- Il Query Optimizer di SQL Server è cost-based
 - Per scegliere quale metodologia utilizzare per costruire il piano di esecuzione si basa su diversi parametri, tra cui le **Statistiche**
- Contengono informazioni sulla distribuzione dei dati
 - Vengono create automaticamente se l'opzione Auto create statistics è ON
 - sulle chiavi degli indici
 - sulle colonne
 - Possono essere create manualmente
 - `CREATE STATISTICS NamePurchase ON AdventureWorks2014.Person.Person (BusinessEntityID, EmailPromotion) WITH FULLSCAN, NORECOMPUTE;`

Statistiche non accurate

- Statistiche non accurate o mancanti possono risultare in piani di esecuzione errati
- Le statistiche vengono aggiornate:
 - Automaticamente se l'opzione Auto Update Statistics del database è ON
 - Ulteriori opzioni: asincrone, incrementali (2016), trace flag 2371 (2012+)
 - Manualmente
 - UPDATE STATISTICS
 - EXEC sp_updatestats
- Includere l'aggiornamento delle statistiche nei piani di manutenzione di SQL Server

Parameter sniffing

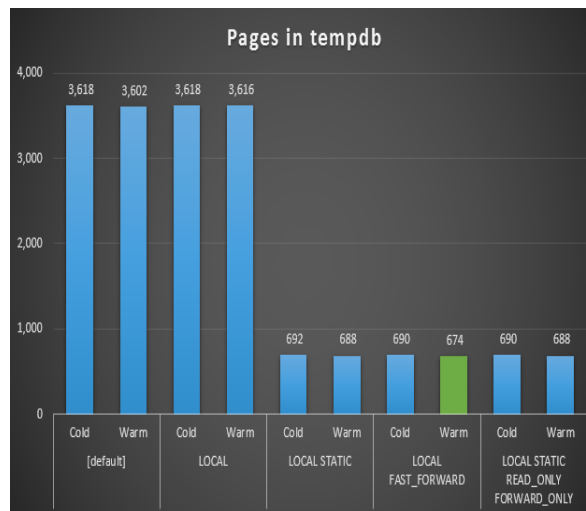
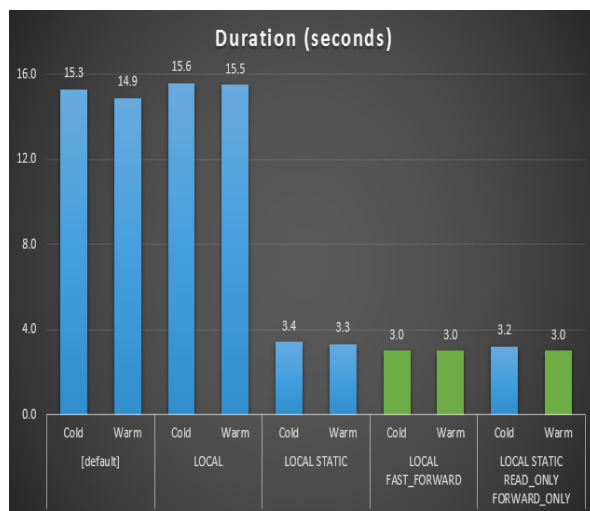
- SQL Server sceglie il piano di esecuzione in funzione dei parametri passati alla stored procedure, memorizzando il piano in cache. Le successive esecuzioni riutilizzano lo stesso piano.
- Ma il piano di esecuzione potrebbe non essere quello giusto se usato con altri parametri
 - Dipende dalla distribuzione dei dati
- Sintomi
 - Performance intermittenti
 - Differenza tra actual e estimated rows nel piano di esecuzione
- Soluzioni
 - OPTIMIZE FOR query hint
 - Ultima spiaggia – RECOMPILE query hint
 - Ultima delle ultime spiagge – Plan Guides
 - Ultimissima spiaggia – spegnere parameter sniffing (trace flag 4136)

Cursori

- SQL Server dà il meglio di sé nella gestione di “set” di dati
- L’approccio procedurale con l’elaborazione iterativa dei dati non è performance
 - Cursori
 - Cicli
 - Ricorsioni
- Restano comunque validi in alcuni scenari dove non è possibile lavorare i dati in modalità set-based

Cursori

- Se proprio dovete usarli...
 - Almeno DECLARE cursor-name CURSOR **LOCAL FAST_FORWARD**



<https://sqlperformance.com/2012/09/t-sql-queries/cursor-options>

Best practices

- SET NOCOUNT ON
- Evitare condizioni di ricerca non SARGable
- Evitare operazioni aritmetiche o funzioni nelle clausole WHERE
- Evitare viste di viste di viste...
- Verificare che non ci siano conversioni implicite
- Eliminare o ridurre l'uso di cursori
- E' necessario includere tutti i campi (SELECT *) ?
- E' necessaria la ORDER BY ?
- Ci sono table variables ? Cambia convertendole in temp tables ?

Tips per i DEV



- DBCC OPTIMIZER_WHATIF
 - Fa credere alla vostra macchina di essere più o meno
- DBCC AUTOPILOT / SET AUTOPILOT ON
 - Crea ed utilizza gli “hypotetical indexes”
 - Utile con database molto grandi per ridurre i tempi di creazione degli indici
- Clone del database “Statistics Only” - <http://bit.ly/2fOSVin>
 - Copia del database con schema e statistiche del database originale
 - Utile per riprodurre e diagnosticare l'esecuzione delle query senza avere i dati
 - Da SQL Server 2014 SP2/2016 SP1 → DBCC CLONEDATABASE
 - <http://bit.ly/2gJdpNQ>

Q&A

Questions?

November 26°, 2016





#sqlsatParma
#sqlsat566

THANKS!

November 26°, 2016

