

Údržba produkčných databáz MSSQL Server

Produkčné databázy prevádzkované na MSSQL Serveri podobne ako iné systémy vyžadujú pravidelnú údržbu a optimalizáciu. Zanedbanie týchto úkonov má často za následok spomalenie systému, zahltenie systémových prostriedkov servera, neoptimálne využívanie optimalizačných nástrojov a v neposlednom rade najmä zhoršenú užívateľskú odozvu pri práci v informačných systémoch. Údržba databáz by mala byť súčasťou dôležitých správcovských úloh, podobne ako pravidelná záloha, kontrola vykonaných záloh, testovacia obnova záloh, monitorovanie systémových prostriedkov servera a pod.

Každá produkčná databáza obsahuje tabuľky a v nich dáta, ku ktorým server pristupuje pri čítacích a zapisovacích operáciách a s ich narastajúcim objemom (počtom záznamov) sa zhoršuje resp. spomaľuje vyhľadávanie a zápis. Hlavnými optimalizačnými prvkami týchto operácií sú indexy a triediace kľúče. Každý správne navrhnutý index je nenahraditeľným nástrojom pri čítacích operáciách, no zároveň vytvára zvýšenú i/o réžiu na serveri pri zápisoch. Nesprávne navrhnutý index má rovnako negatívny vplyv na výkon servera ako nevyužívaný index, preto je dôležité sledovať aj ako sa jednotlivé indexy používajú jednak z pohľadu celkového používania, ale aj pri operáciách typu seek, scan a update. To že sa index od posledného štartu servera nepoužil však ešte neznamená, že nie je v databáze potrebný a že je ho možné vypnúť alebo zmazať, pretože ak sa niektoré operácie v systéme nevykonávajú na každodennej báze, ale povedzme len raz mesačne, tak ich spustením môžu znova SQL serveru chýbať. Štatistiky o používaní alebo nepoužívaní indexov SQL serverom sú relevantné len od posledného štartu jeho služby vo Windows. K týmto údajom je možné pristupovať pomocou dynamic management views, ktoré sú súčasťou SQL servera od verzie 2005 a vyššie.

- ❖ sys.dm_db_index_usage_stats
- ❖ sys.dm_db_index_operational_stats
- ❖ sys.dm_db_missing_index_groups
- ❖ sys.dm_db_missing_index_group_stats
- ❖ sys.dm_db_missing_index_details

Z hľadiska databázy je za aktuálnosť zodpovedné nastavenie „automatická aktualizácia štatistík“. Toto nastavenie udržiava relevantné informácie a štatistiky o každom indexe pri ktorejkoľvek operácii s dátami.

SQL Server si eviduje štatistiku informácií o každom indexe a túto následne využíva pri rozhodovaní, ktorý z nich využije pre optimálne vykonanie danej operácie. Ak index chýba, alebo nie je navrhnutý optimálne pre určitú operáciu môže sa SQL governor rozhodnúť nepoužiť žiadny z dostupných indexov – nastane table scan, alebo použije index, ktorý len čiastočne pokrýva požiadavku na výber dát a nastane indexscan. Ideálny stav je potom index seek, pri ktorom sa nájde ideálny index a teda sa daná operácia vykonáva optimálne. V zložitejších systémoch nie je možné dosiahnuť ideálny stav pri všetkých operáciách, pretože každý select môže vo výbere, podmienke, joine a triedení zohľadniť iný zoznam alebo počet atribútov a navrhnuť tak ideálny index preto nie je možné. Veľmi dôležitá je tak podmienka where, ktorá z daného výberu záznamov eliminuje vysoký počet záznamov a znižuje tak množinu dát s ktorou sa pracuje. Zároveň platí, že s každým novým nonclustered indexom narastá násobne veľkosť tabuľky o celkovú kapacitu dát obsiahnutých v tabuľke. Dáta v týchto indexoch sú online aktualizované, preto každý zápis do tabuľky predstavuje zvýšenú réžiu servera pri zápise a usporiadaní dát v každom nonclustrovanom indexe (kópií dát). Jedinú výnimku tvorí clustered index, ktorý môže existovať v tabuľke iba jeden, pretože tento typ dáta usporiada priamo na úrovni tabuľky (nie kópie).

Informácie o štatistikách môžu byť ale nepresné napr. pri veľkých objemoch vkladanych alebo prepisovaných dát na veľmi intenzívne používaných tabuľkách, alebo pri mazaní tabuliek pomocou príkazu `truncate table`. Na takýchto tabuľkách je vhodné spustiť aspoň raz denne aktualizáciu štatistik. Neaktuálnosť týchto údajov má za následok neoptimálne rozhodovanie SQL governora, t.j. že sa rozhodne nevyužiť dostupný index. V niektorých prípadoch sme sa stretli s používaním syntaxe „using index“ v skriptoch, používanie tejto syntaxe nepovažujeme za správne, pretože časom môže programátor napísať iný vhodnejší index, ktorý bude touto syntaxou potlačený a server sa preň nemôže rozhodnúť samostatne na základe štatistiky, hoci ideálny index pre príkaz existuje.

V praxi sa môžeme stretnúť s dvoma príkazmi na aktualizáciu štatistiky:

`DBCC UPDATEUSAGE` – koriguje záznamy o využití, rezervácií a stránkach atribútov každej tabuľky v tabuľke `sysindexes`, neaktualizuje však záznamy o veľkostiach / kapacitách neclusterovaných indexov.

`EXEC sp_updatestats` – tento príkaz aktualizuje štatistiky a je presnejší ako voľba nastavená na úrovni databázy - „automatická aktualizácia štatistik“ V niektorých prípadoch sa odporúča automatickú aktualizáciu vypnúť a vykonať ju týmto príkazom manuálne minimálne každý deň.

Tento príkaz je vhodné spúšťať najmä ak dochádza k výkonnostným problémom na SQL serveri a je to v zásade prvý opravný krok pred hlbšou analýzou problémov na úrovni príkazu, indexov a pod. Je to najmä z dôvodu že mnoho problémov s výkonnosťou queries je spôsobených práve neaktuálnosťou štatistických údajov. Tento krok môže výrazne ušetriť čas analýzy výkonnostného problému.

Reálne sa týmto príkazom aktualizujú údaje o tabuľkách:

- počte data pages využívaných každou tabuľkou;
- počte záznamov v každej tabuľke;
- počte insertov, updatov, deletov ktoré majú dopad na kľúče tabuliek od poslednej aktualizácie štatistiky;

Okrem toho sa aktualizujú i štatistiky o indexoch:

- histogram o distribúcií dát cez prvý (kľúčový) atribút každého indexu
- priepustnosť atribútových prefixov
- priemerná dĺžka kľúča každého indexu

SQL Server môže evidovať štatistiku aj o ktoromkoľvek atribúte, ktorý programátor v príkaze špecifikuje. Všetky tieto informácie slúžia Query optimizéru správne sa rozhodovať ako najlepšie, efektívne a najjednoduchšie vykonať príkaz, alebo proces, ktorý informačný systém na serveri spúšťa.

V prípade automatickej aktualizácie štatistik povolenej na úrovni databázy (predvolené nastavenie) sa štatistiky aktualizujú automaticky pri reindexácii – defragmentácii indexov, čo je z pohľadu výkonnosti ďalším z optimalizačných nástrojov ovplyvňujúcich výkon SQL servera.

Pri veľmi intenzívnej réžii zápisov a updatov do konkrétnej tabuľky server ale nemusí stíhať dáta v indexoch rýchlo usporiadať - aktualizovať štatistiku, nastáva ich fragmentácia a zahltenie systému touto réziou. V takýchto prípadoch je lepšie automatické obnovenie štatistiky odstaviť a zároveň naplánovať ručnú aktualizáciu minimálne jedenkrát denne v čase nízkej prevádzkovej réžie. V žiadnom prípade však neodporúčame nevykonávať žiadnu aktualizáciu, to je ten najhorší scenár.

Defragmentácia indexov je proces z pohľadu systémových nárokov na server veľmi intenzívny a nemal by sa spúšťať voči všetkým indexom v databáze. Najmä pri objemných databázach sa jedná o veľmi vyťažujúci a časovo náročný proces pre hardvér servera. Našťastie aj v tomto prípade si SQL server generuje štatistiku fragmentácie

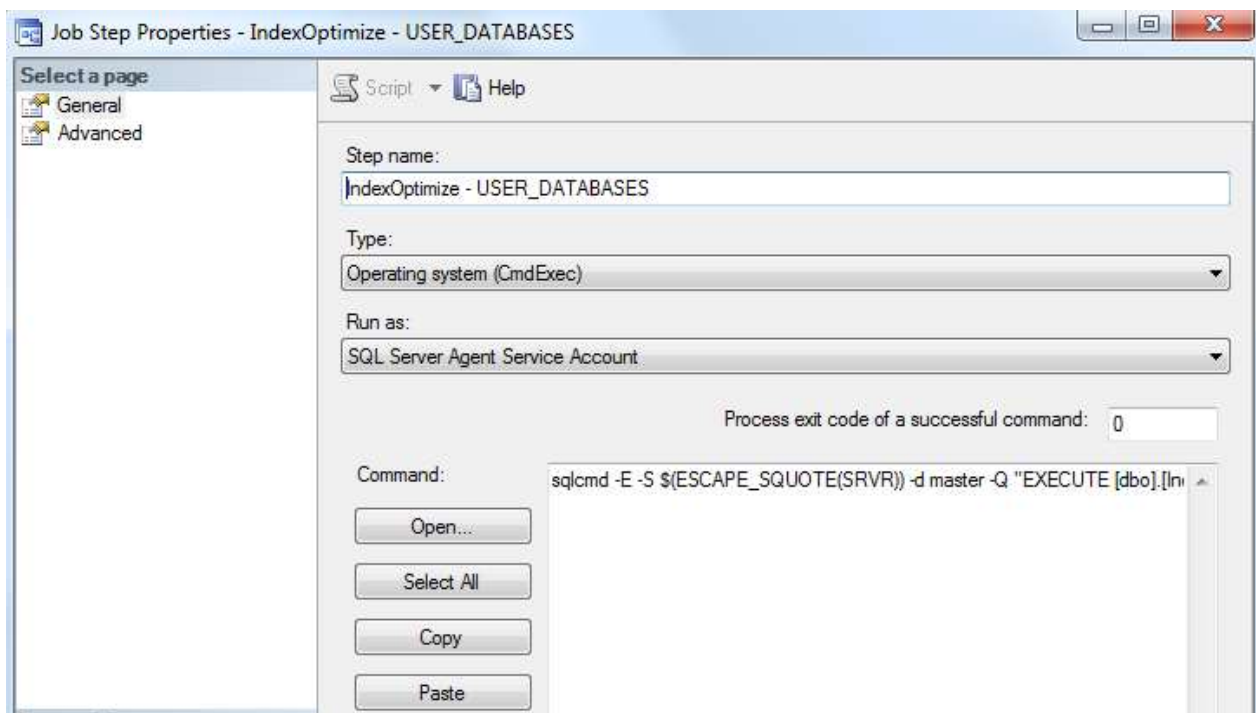
indexov a podľa nej je možné správcom naplánovať reindexáciu takých indexov, ktoré dosahujú vyšší stupeň fragmentácie, najmä na často sa meniacich tabuľkách s väčším počtom indexov a objemom dát minimálne na týždennej báze.

Oba procesy – aktualizácie štatistík a defragmentáciu je možné vykonávať v rámci plánu údržby, ktorý je možné automatizovať cez SQL Server Agentu ako plánovanú úlohu aj s emailovou notifikáciou pre prípad, že sa niektorý z krokov nepodarí vykonať z dôvodu chyby. Správca bude o chybe vždy informovaný.

Pre jednoduchosť dávame do pozornosti údržbový skript od Ola Hallengren, ktorý vykoná inštaláciu potrebných procedúr a logovacích tabuliek ako aj zavedie sadu jobov do SQL Agentu. Viacero zdrojov na stránkach Microsoftu odporúča túto sadu využívať namiesto bežne známych spôsobov údržby.

<https://ola.hallengren.com/>

Celá sada nástrojov sa inštaluje pomocou MaintenanceSolution.sql skriptu z vyššie uvedenej stránky. Vzniknuté joby sú v SQL Agentovi pripravené na spustenie, ale je ich ešte potrebné naplánovať, samé sa totiž nenastavia.



Jednotlivé plány spúšťajú nainštalované uložené procedúry parametricky. Každý job je možné teda parametricky upraviť podľa želaného správania.

Napr. ak požadujeme aktualizovať štatistiky užívateľských databáz, upravíme volanie procedúry dbo.IndexOptimize a požadovanú vlastnosť do príkazu doplníme.

```
EXECUTE dbo.IndexOptimize @Databases = 'USER_DATABASES',
@FragmentationLow = NULL,
@FragmentationMedium =
'INDEX_REORGANIZE,INDEX_REBUILD_ONLINE,INDEX_REBUILD_OFFLINE',
@FragmentationHigh = 'INDEX_REBUILD_ONLINE,INDEX_REBUILD_OFFLINE',
@FragmentationLevel1 = 5,
@FragmentationLevel2 = 30,
@UpdateStatistics = 'ALL',
@OnlyModifiedStatistics = 'Y'
```

Pre podrobnejší výpis možností nastavenia a detaily o každom parametri vid' <https://ola.hallengren.com/sql-server-index-and-statistics-maintenance.html> na konci stránky sú uvedené aj príklady použitia pre konkrétne situácie

Históriu vykonaných operácií a ich výsledky je možné sledovať selectom do logovacej tabuľky:

```
select * from master.dbo.CommandLog order by ID desc
```

Joby ktoré správca nevyužije, môže z SQL Agentu vymazať.

Pre účely tejto dokumentácie nebudeme uvádzať konkrétne odporúčania, t.j. ktoré parametre oproti predvoleným nastaveniam jobov meniť. K úpravám skriptov - volaní jobov je potrebné pristupovať s ohľadom na reálnu prevádzku a brať pri tom ohľad aj na obmedzenia a možnosti každej inštalácie. Je to ale výborný nástroj ako udržiavať databázy a dáta v dobrej kondícii, rovnako ako aj hľadať a riešiť vzniknuté problémy.

V Bratislave dňa 18.11.2016

INTELSOFT