

Defragmentace indexů



ESO9 international a.s.

U Mlýna 22
141 00, Praha

..... www.eso9.cz

Defragmentace indexů	3
Trocha teorie	3
Optimalizace Fill factoru v indexech.....	4
Vliv na velikost databáze	5
Postup nasazení.....	6
Nastavení JOBu	7
Parametry aplikace (9.6.1).....	7
Skupina Defragmentace	7

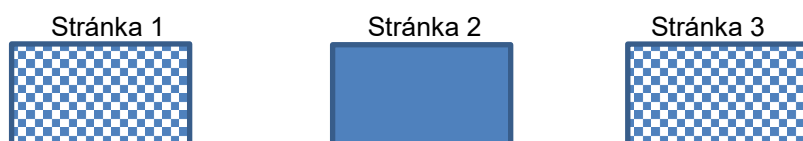
Defragmentace indexů

Trocha teorie

Základní jednotkou paměti, se kterou pracuje SQL server, je takzvaná stránka. To je úsek paměti o velikosti 8 kB - pokud jsou potřeba načíst data z této stránky (a to jak vlastní data, nebo index), SQL server musí načíst (zapsat) vždy celou tuto stránku.

Při tvorbě indexu je možno zadat hodnotu FILLFACTOR - to je hodnota (od 1 do 100), která určuje procento prostoru na každé databázové stránce, které má být vyplněno daty. Zbytek rezervuje na každé stránce jako volné místo pro budoucí naplnění. Například zadání hodnoty faktoru výplně 80 znamená, že 20 procent každé stránky zůstane prázdné, což poskytne prostor pro rozšíření indexu, když se data přidají do tabulky. Výchozí hodnota faktoru (pokud to při založení indexu není specifikováno) je 100%, což znamená, že stránky jsou naplněny na plnou kapacitu.

V indexových stránkách jsou hodnoty v indexu seřazeny dle definovaného třídění, při vložení nového záznamu je tedy nutno hodnotu indexu uložit na konkrétní místo v konkrétní stránce.



V případě Stránky 1, která není zcela zaplněna, dojde pouze k vložení nové hodnoty indexu, případně se provede přeskupení hodnot v rámci této stránky. Když je však potřeba doplnit hodnotu indexu na plnou stránku (Stránka 2), kde již není volné místo, databázový mechanismus ponechá přibližně polovinu řádků na původní stránce a druhou polovinu přesune na novou stránku. Je to proto, aby tam bylo místo pro záznam nové hodnoty indexu. Tato reorganizace se nazývá rozdělení stránky.



Rozdělení stránky vytváří prostor pro nové záznamy. Provedení této operace však může nějakou dobu trvat a je to operace náročná na zdroje (alokace prostoru na disku, ...). Stránky 2A a 2B neleží na disku „vedle sebe“, takže přístup i na „blízké“ hodnoty indexu může znamenat přístup na různá místa diskového prostoru (to může být optimalizováno teprve až při rebuildu indexu).

Správně vybraná hodnota Fill factoru může snížit potenciální rozdělování stránek tím, že poskytuje dostatek místa při vkládání nových záznamů do tabulky.

- Nižší hodnota Fill factoru může tedy četnost požadavků na rozdělování stránek snížit, ale naopak bude vyžadovat více úložného prostoru a tím může snížit výkon čtení.
- Například hodnota Fill factoru 50 způsobí, že se výkon čtení konkrétní tabulky (indexu) může snížit až dvakrát. Výkon čtení je snížen, protože index obsahuje 2 x více stránek, a proto zvyšuje počet IO operací disku potřebných k načtení dat.
- Zvolit vhodnou hodnotu Fill factoru není jednoduché, takže způsob „Nastaví se to všude třeba na 80“, nemusí být vůbec optimální!

Fragmentace je závislá na mnoha faktorech. Závisí na tom, jak často a kolik je vkládáno nových záznamů, jakým způsobem je tvořen index apod.

- Například index, který je tvořen sloupcem IDENTITY, by měl záznamy v indexu logicky přidávat pouze na konec indexu, tedy by k defragmentaci dle této teorie nemělo docházet.
- To však ale nemusí být vždy pravda, protože v případě primárního klíče jsou zde kromě hodnot klíče uložena i data celého záznamu, a při aktualizaci například položek typu varchar (když dojde ke zvětšení jejich délky) bude rovněž docházet k fragmentaci.

Optimalizace Fill factoru v indexech

Jak tedy „správně“ určit hodnotu Fill factoru ?

Předpřipravený algoritmus pracuje tak, že opakovaně (například každý den v noci) zjistí, které indexy jsou nejvíce defragmentované. U nich provede snížení hodnoty Fill factoru a provede reindexaci. Následující dny se proces stále opakuje až do té doby, kdy je defragmentace indexu menší než nějaká definovaná hodnota.

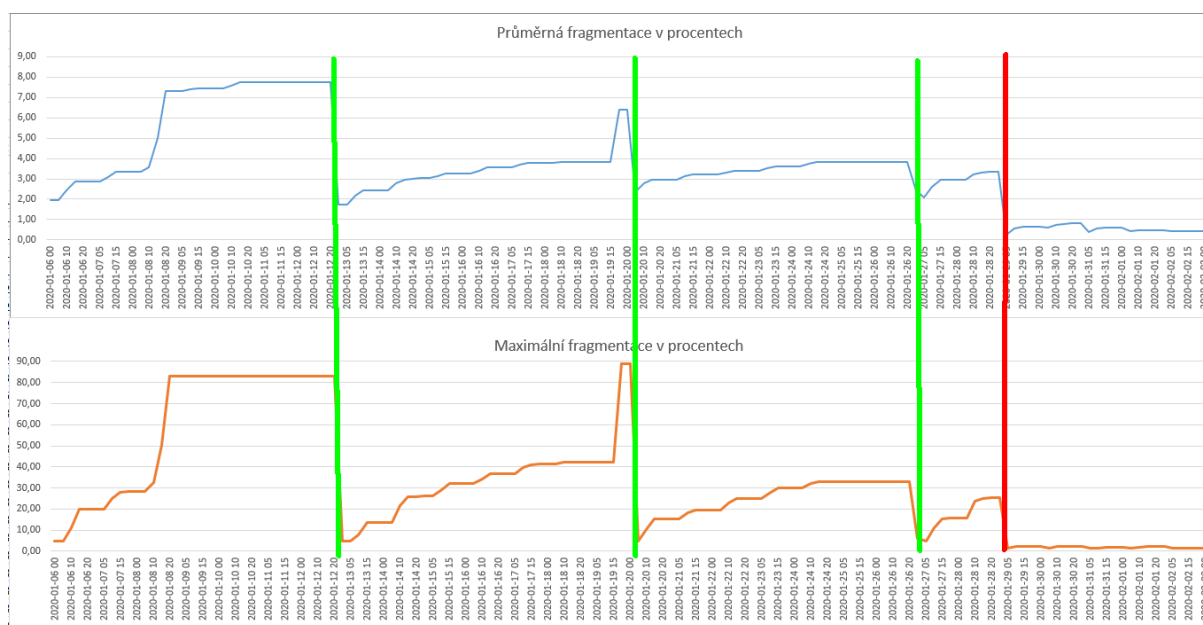
Níže uvedený příklad popisuje, jaký vliv mělo provádění optimalizace Fill factoru na konkrétní databázi (DB). Jednalo se o skutečnou produkční databázi, celková velikost databáze byla 5,2 GB (1,2 space available). Optimalizace se prováděla pro všechny indexy s počtem stránek větších než 1000 a požadovanou maximální defragmentací 1,3%.

Tabulky s největším počtem řádků (v milionech), v závorce je velikost v % z celkové velikosti DB (těchto 7 tabulek tvoří 87% velikosti DB):

- SDOK – 1,86 (46%)
- SDOKAdd – 1,44 (9%)
- HisSdok – 0,91 (4%)
- UcetZap – 0,79 (11%)
- ZbozSklSarze – 0,47 (3%)
- ZbozSklSarzeADD – 0,46 (1%)
- HDOK – 0,30 (13%)

V databázi je celkem 44 indexů, jejichž velikost je větší než 1000 stránek. Pro tyto indexy byl pravidelně 5x denně prováděn zápis do statistiky, ve které se zaznamenávalo procento defragmentace. Následující 2 grafy ukazují, jak se vyvíjela průměrná a maximální fragmentace na těchto 44 indexech od 6.1.2020 do 3.2.2020.

- Svislé zelené čáry zobrazují okamžiky dosud prováděných pravidelných týdenních reindexací (neděle večer)
- Modré a oranžové křivky zobrazují průměrné, resp. maximální procenta defragmentace. Tato se zvyšovala až do další týdenní reindexace, na některých tabulkách až nad 80 % (na tabulkách byl nastaven stále Fill factor 100%).
- Červená čára zobrazuje okamžik počátku nasazení optimalizace Fill factoru (ve středu 29.1.2020).
- Po 6 dnech se průměrná defragmentace sledovaných 44 indexů ustálila na hodnotě 0,5 (maximální je asi 2,2)

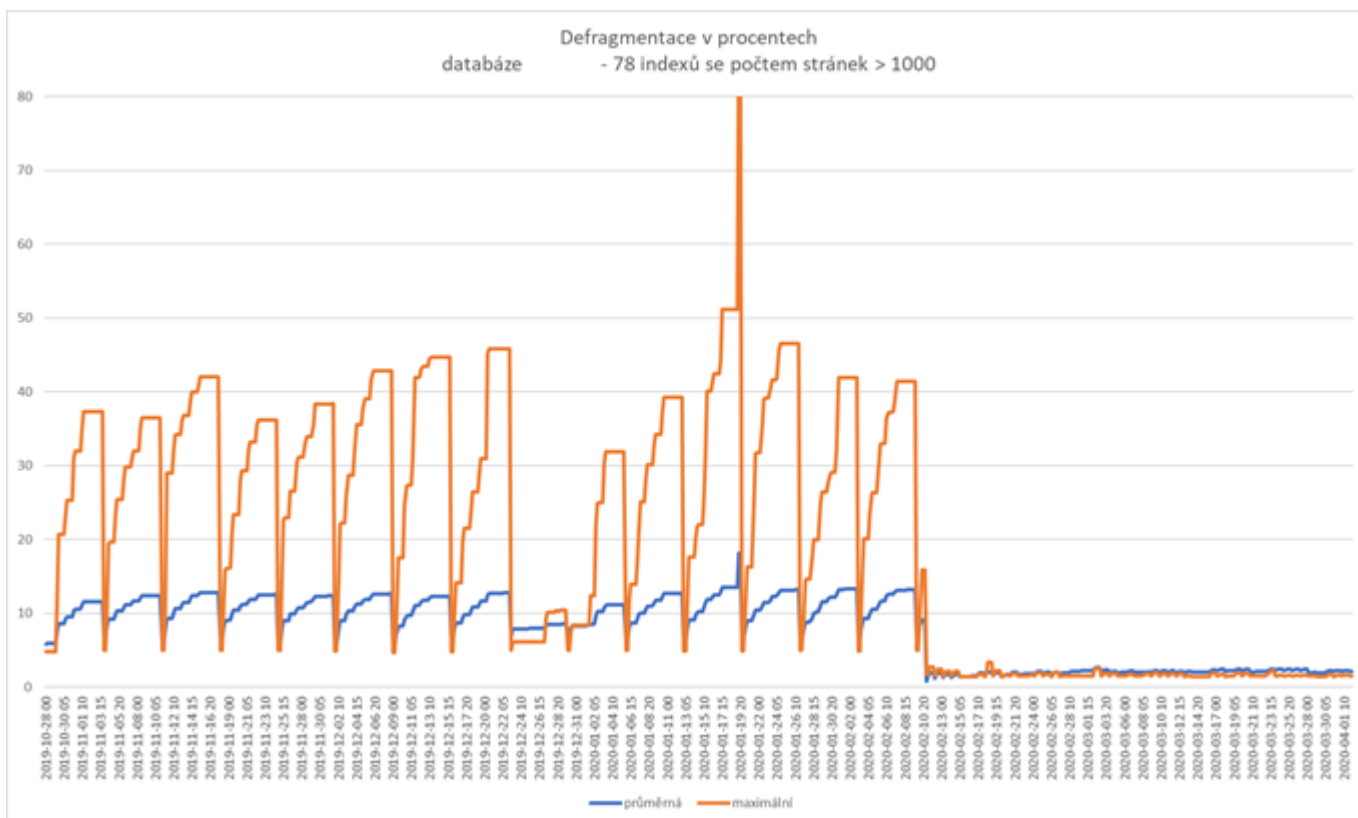


Vliv na velikost databáze

	Velikost - data	Velikost - indexy	Počet řádků
Před optimalizací	2,46 GB	1,22 GB	7,763 mil
Po 8 dnech	2,51 GB (+42 MB)	1,20 GB (-18 MB)	7,792 mil (+29 tis)

Přestože za 8 dní provozu přibýlo v tabulkách asi 42 MB dat, velikost indexů se po optimalizaci Fill factoru paradoxně zmenšila. Má to logické vysvětlení – silně fragmentovaný index s Fill factorem 100% zabírá více místa než málo fragmentovaný index s nižším faktorem (např. 95).

Následuje příklad další databáze – kdy došlo k optimalizaci Fill factoru, je z grafu zřejmé:



Postup nasazení

Od verze 6.1 jsou standardně v produkčních databázích (typu Start i PAM) tyto nové tabulky a procedury:

Tabulka **AgentIndexRebuilds**

- obsahuje podklady o indexech, na základě kterých se rozhoduje o nastavení Fill factoru (počet stránek, záznamů, hodnoty fragmentace, počty rozdělených stránek apod.)

Tabulka **DefragLog**

- obsahuje informace o provedení rebuild/reindexace indexu

Procedura **spResetTrackPageSplits**

- slouží pro vytvoření takzvané event_sessions, která sleduje rozdělování stránek (používá se v proceduře spServer_SelfTunningFillFactor)

Procedura **spLogDefrag**

- pomocná procedura pro zápis do tabulky DefragLog

Procedura **spSelfTunningFillFactor**

- vlastní procedura pro vyhodnocení a nastavení Fill factoru. Procedura má parametry:
 - @TopWorkCount - definuje maximální počet indexů, pro které se bude provádět nastavení Fill factoru a následná reindexace. Defaultní hodnota = 30 (tedy optimalizace se v jednom chodu provede pro 30 nejvíce defragmentovaných indexů).
 - @MinFragmentation - do zpracování se budou brát pouze indexy, které mají defragmentaci vyšší než zde zadaná hodnota. Defaultní hodnota = 2.5, což znamená, že se bude provádět optimalizace pouze pro indexy, jejichž defragmentace je větší než 2.5%. Minimální možná hodnota je 1.2 (pokud je zadaná nižší hodnota, použije se hodnota 1.2).
 - @MinPage - do zpracování se budou brát pouze indexy, které mají počet databázových stránek vyšší než zde zadaná hodnota. Defaultní hodnota = 50, což znamená, že se bude provádět optimalizace pouze pro indexy, jejichž velikost je větší než 50 stránek. Minimální možná hodnota je 20 (pokud je zadaná nižší hodnota, použije se hodnota 20)

Procedura **spServer_SelfTunningFillFactor**

- provádí volání procedury spSelfTunningFillFactor ve všech databázích na serveru, ve kterých nalezne proceduru spSelfTunningFillFactor. Má stejné parametry jako procedura spSelfTunningFillFactor, navíc má:
 - @VyberDB – seznam databází oddělených středníkem, pro které se má provádět. Pokud je zadán seznam databází, bude se provádět pouze pro tyto DB, pokud není zadán, bude se provádět pro všechny databáze na serveru
 - @KromeDB - pokud je zadán seznam databází, tak se pro tyto nebude provádět
- Mohou být zadány oba dva parametry, je možno využít konvence „LIKE“, například:
@VyberDB = 'ESO%', @KromeDB = '%TEST;%PAM'
 - V takovém případě se bude provádět pro všechny databáze, jejichž název začíná ESO a současně na konci nemá TEST nebo PAM.

POZOR! V případě, že neomezíte výběr databází pomocí těchto parametrů, bude se provádět **ve všech databázích** typu ESO9 a ESO9PAM verze 6.1 a vyšší na daném SQL serveru. Případné vyloučení z provádění optimalizace lze rovněž nastavit parametrem **Defrag_Auto** (viz popis parametrů níže). Standardní hodnota tohoto parametru je po instalaci verze 6.1 nastavena na hodnotu 1 (= optimalizace Fill factoru je povolena).

Výše uvedené tabulky a procedury jsou ve verzovém scriptu zakládány pouze pro vlastní produkční databáze, nikoliv pro LOG a DOC databáze. V případě potřeby zprovoznění optimalizace Fill factoru na LOG a DOC databázích je k dispozici SQL script **TunningFillFactor.sql**, kterým je možno tyto tabulky a procedury založit i v databázích LOG a DOC. Není možno zde ale založit parametry – hodnoty pro provádění optimalizace Fill factoru. Vždy se tedy uplatní hodnoty zadané v parametrech při volání procedury (popis parametrů viz níže).

Nastavení JOBu

Vlastní provádění nastavení Fill factoru by mělo být prováděno JOBem, který by měl být spouštěn denně v době, kdy se předpokládá minimální přístup k databázi. Po dobu rebuildu/reindexace indexu není tento index dostupný, v závislosti na velikosti tabulky tedy může být po tuto dobu nedostupný.

Upozornění:

- Pokud již máte nastaven nějaký JOB pro provádění reindexace / rebuildu indexů, **vypněte ho**.
- Pokud v údržbách používáte DBCC SHRINKFILE, je nutno ho používat s parametrem TRUNCATEONLY, jinak po jeho použití dojde k opětovné defragmentaci indexů!

Více o této problematice např. na <https://www.mssqltips.com/sqlservertip/4368/execute-sql-server-dbcc-shrinkfile-without-causing-index-fragmentation/>

Příklad JOBu pro spuštění nad jednou konkrétní databází:

```
exec spSelfTunningFillFactor
@TopWorkCount = 30,
@MinFragmentation = 2.5,
@MinPage = 50
GO
```

Příklad JOBu pro spuštění nad všemi databázemi SQL serveru: (je třeba spouštět nad jednou z databází, která tuto proceduru obsahuje)

```
exec spServer_SelfTunningFillFactor
@TopWorkCount = 20,
@MinFragmentation = 5.0,
@MinPage = 200,
@VyberDB = "",
@KromeDB = 'ESO9START%;%TEST%'
GO
```

Doporučení:

Pokud chcete provádět optimalizaci nad více databázemi, pak ji neprovádějte opakovaným spouštěním procedur spSelfTunningFillFactor pro jednotlivé databáze, ale hromadně pomocí procedury spServer_SelfTunningFillFactor. Důvodem je to, že optimalizace Fill factoru se provádí jednak na základě defragmentace, ale také dle počtu rozdělených stránek – a tento počet se zjišťuje na základě event_sessions, která shromažďuje data pro celý SQL server (jsou společná pro všechny databáze) a po zpracování je nutno ji resetovat (zastavit a spustit - viz procedura spResetTrackPageSplits).

V závislosti na počtu databází a jejich velikostí je nutno zvážit, na kolika databázích a s jakými parametry optimalizaci provádět (a tedy kolik a jakých indexů reindexovat) – od toho se samozřejmě bude odvíjet doba provádění. Zpočátku doporučujeme provádění jen pro několik největších indexů s velkou fragmentací – tedy například s parametry: max. 20 indexů, minimální fragmentaci 5%, minimální počet stránek 1000, a po čase změnit například na (30 / 2.0 / 50).

Parametry aplikace (9.6.1)

Skupina Defragmentace

Aby bylo možné v případě potřeby pro některou aplikaci nastavit odlišné hodnoty parametrů optimalizace Fill factoru, existuje v aplikaci skupina parametrů „Defragmentace“, v níž jsou k dispozici 4 parametry. Pomocí nich lze nastavit hodnoty, se kterými se má provádět procedura spSelfTunningFillFactor, případně lze její provádění zakázat.

Defrag_Auto – Povolení pro provádění optimalizaci fill-factoru

- Definuje, zda je pro databázi povoleno provádění optimalizace Fill factoru (0 = zakázáno, 1 = povoleno).
 - Pokud je procedura spSelfTunningFillFactor spuštěna, a hodnota parametru je nastavena na hodnotu 0, pak se optimalizace Fill factoru v této databázi neprovede (procedura končí).
 - Standardní hodnota parametru je nastavena na hodnotu 1=povoleno.

Pro následující 3 parametry platí, že pokud parametr v aplikaci není vyplněn, uplatní se parametr, se kterým se volá procedura spSelfTunningFillFactor. Pokud je některý parametr vyplněn, pak má přednost před parametrem, se kterým je volána procedura spSelfTunningFillFactor.

..... www.eso9.cz

Při instalaci verze 6.1 se tyto parametry založí bez hodnot (jsou prázdné):

Defrag_MaxPocIndexu – Počet indexů při optimalizaci fill-faktoru

- Zadáva se zde maximální počet indexů, pro které se má v jednom chodu (spuštění optimalizační procedury spSelfTunningFillFactor) provádět nastavení Fill factoru a následná reindexace.

Defrag_MinPage – Minimální velikost při optimalizaci fill-faktoru

- Zadáva se zde minimální velikost indexu (počet databázových stránek), pro které se má provádět nastavení Fill factoru a následná reindexace.

Defrag_MinFrag – Minimální fragmentace při optimalizaci fill-faktoru

- Zadáva se zde minimální hodnota fragmentace indexu, pro kterou se má provádět nastavení Fill factoru a následná reindexace

Doporučení: parametry Defrag_MaxPocIndexu, Defrag_MinPage a Defrag_MinFrag nevyplňovat a ponechat v režii spuštění procedury.

Poznámka: protože DOC a LOG databáze parametry neobsahují, tak nad těmito databázemi se vždy použijí parametry zadané ve volání procedury.