Partitioning v Microsoft SQL Serveru

RNDr. David Gešvindr, Ph.D.
MVP: Data Platform | MCSE: Data Platform | MCT
david@wug.cz
@gesvindr

Osnova

1. **Představení principů partitioningu**
2. Nasazení partitioningu pro nové i existující tabulky
3. Operace s partitions
4. Problémy, které nám pomáhá partitioning řešit

Uložení řádků v tabulce

- Microsoft SQL Server ukládá řádky tabulky zakódované do 8 kB datových stránek
- Datové stránky tabulky jsou organizovány buď jako datové struktura **halda (heap)** nebo tvoří **clustered index (B+ strom)**
- Při zakládání tabulky (indexu) vybíráme, ve které filegroup budou uloženy datové stránky
- Filegroup je tvořena jedním či více datovými soubory, kde jsou uloženy datové stránky
  - U filegroup s více datovými soubory, se datové stránky alokují podle podílu volného místa

Partitioning

- Partitioning umožňuje rozdělit velkou tabulku z pohledu uložení a správy na více menších skupin řádků (partitions)
  - Pro klienta se stále chová tabulka jako celek a partitions nejsou viditelné
- Každá partition je tvořena nezávislým B+ stromem či haldou
  - Izolace údržby indexu jen na podmnožinu řádků tabulky
- U každé partition si vybíráme, ve které filegroup je uložena
  - Rozdělení řádků jedné tabulky podle stáří na různě výkonné disky

Definice hranic partitions

- Hranice jednotlivých partitions jsou určeny **partitioning funkcí**
- U této funkce si vybíráme, jestli má pracovat v režimu **range left** nebo **range right**
  - V režimu **range left** je hraniční hodnota součástí levé partition jako maximální hodnota
  - V režimu **range right** je hraniční hodnota součástí pravé partition jako minimální hodnota
  - Pro datový typ datum je výhodnější používat range right a zadávat jako hranice začátek dne / měsíce / roku

Mapování partitions na FileGroups

- Partitions definované pomocí partitioning funkce musíme namapovat na filegroups v dané databázi, kam mají být uloženy
- O toto mapování se stará **partitioning schéma**, které definujeme nad partitioning funkcí
- Partitioning schéma definuje seznam filegroups, kam mají být uloženy partitions určené partitioning funkcí
  - Je možné definovat více filegroups, než je třeba
  - Definujeme také NEXT USED filegroup, která se využije při rozdělování partitions

Výběr sloupce pro rozdělení dat

- Při vytváření tabulky vybíráme **partition klíč** – jeden sloupec, podle kterého se budou rozdělovat řádky na základě partitioning schématu
- Má-li tabulka clustered index, tak tento sloupec musí být součástí klíče clustered indexu
  - Tento sloupec musí být součástí klíče všech jedinečných non-clustered indexů
- Doporučuje se vybrat sloupec, který se téměř vždy vyskytuje v podmínce dotazů
  - SQL Server mechanizmem **partition elimination** umožní rovnou načítat jen data vybrané partition
  - Na tabulkách faktů se často vybírá časová dimenze

Osnova

1. Představení principů partitioningu
2. **Nasazení partitioningu pro nové i existující tabulky**
3. Operace s partitions
4. Problémy, které nám pomáhá partitioning řešit

Vytvoření tabulky využívající partitioning

- Pokud vytváříme novou tabulku využívající partitioning, tak uvedeme místo filegroup, kde má být vytvořena (nebo její clustered index) partition schéma a partition klíč

```
CREATE TABLE dbo.Orders (
[OrderDate] [date] NOT NULL,
[OrderID] [bigint] IDENTITY(1,1) NOT NULL,
[CustomerID] [int] NULL,
[TotalDue] [decimal](19, 4) NULL,
CONSTRAINT [PK_Orders] PRIMARY KEY CLUSTERED
  ([OrderDate] ASC, [OrderID] ASC))
ON psOrders(OrderDate);
```

Aplikace partitioningu na existující tabulku

- Pro aplikaci partitioningu na existující tabulku i s uloženými daty je nutné znovu založit její clustered index nad partition schématem
- Existující indexy nad tabulkou mohou být také znovu založeny nad stejným partition schématem se stejným partition klíčem a potom jsou označovány jako **partition aligned indexy**
- SQL Server Management Studio obsahuje průvodce, který vygeneruje potřebný kód

Osnova

1. Představení principů partitioningu
2. Nasazení partitioningu pro nové i existující tabulky
3. **Operace s partitions**
4. Problémy, které nám pomáhá partitioning řešit

Operace s partitions

- Vytvořenou partitioning funkci a partitioning schéma lze modifikovat i pokud má tabulka již uložená data
- Podporované operace jsou:
  - Partition Split
  - Partition Merge
  - Partition Switch

Partition Split

- Stávající interval definovaný v partition funkci je možné rozdělit na dva intervaly
  - Pokud je partition funkce použita u více tabulek, dojde k rozdělení partition ve všech dotčených tabulkách v jedné transakci
  - Rozdělení partition je off-line operace, je proto dobré minimalizovat dobu nedostupnosti a rozdělovat prázdné partitions
  - Pokud by mělo dojít k zásadní změně rozložení partitions v tabulce, je lepší data okopírovat do nové tabulky nebo založit znovu clustered index
- Partition schema musí definovat NEXT USED filegroup, jinak příkaz selže chybou

Partition Merge

- Je možné spojit dvě sousední partitions do jedné partition úpravou partition funkce
  - SQL Server má limit, kdy tabulka může mít maximálně 15 000 partitions
- Při rušení se musí uvést přesně hraniční hodnota definovaná v partition funkci
- Nově vytvořená partition bude existovat ve filegroup, která původně neobsahovala hraniční hodnotu

Partition Switch

- Je možné vyměnit celé partitions i s daty mezi tabulkami
  - Jedna z tabulek nemusí aktuálně využívat partitioning
- Zdrojová a cílová tabulka musí mít stejné sloupce, indexy a používat stejný sloupec pro partitioning
- Obě tabulky musí být ve stejné filegroup
  - Operace partition switch je prakticky instantní, protože mění pouze metadata a nikdy nekopíruje žádná data
- Cílová partition či tabulka musí být prázdná

Osnova

1. Představení principů partitioningu
2. Nasazení partitioningu pro nové i existující tabulky
3. Operace s partitions
4. **Problémy, které nám pomáhá partitioning řešit**

Zjednodušení údržby velkých indexů

- Některé typy modifikací dat nad tabulkou vedou ke zvýšení fragmentace indexů
- Fragmentace je 2 druhů:
  - Interní fragmentace – Datové stránky nejsou zaplněné tak, jak by mohly
  - Externí fragmentace – Datové stránky nejsou uloženy za sebou v souboru
- Údržba indexu probíhá na úrovni celého indexu a zahrnuje 2 možné operace údržby:
  - Rebuild indexu
  - Reorganizaci indexu

Zjednodušení údržby velkých indexů

- Díky partitioningu jsou indexy nad tabulkou rozděleny na více menších indexů a může dojít k tomu, že se vznikající fragmentace izoluje jen na vybrané partitions, které následně vyžadují údržbu
- Je možné detekovat míru fragmentace jednotlivých partitions indexu
- Operace údržby je možné spouštět jen nad vybranými partitions
  - Nedochází ke zbytečné údržbě částí tabulek, které údržbu nepotřebují
  - Značná úspora času i zátěže spojené s údržbou indexů

Zjednodušení přepočtu statistik

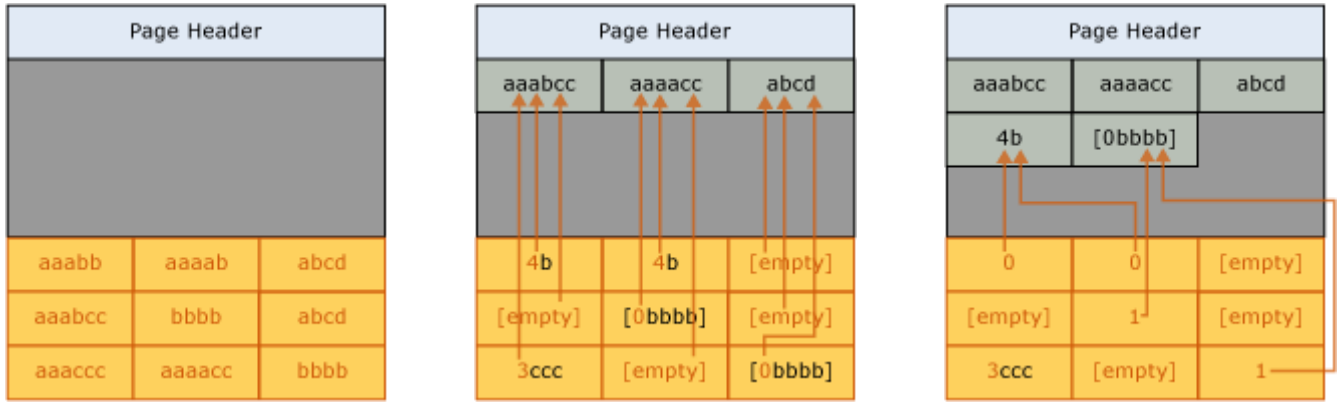
- Od SQL Serveru 2014 je možné na úrovni databáze povolit vytváření **inkrementálních statistik**
- SQL Server vytvoří statistiky pro celou tabulku, ale také navíc pro každou partition
- Při změně dat ve vybraných partitions bude možné nechat přepočítat pouze statistiky daných partitions
  - Neodchází tak zbytečně ke čtení velikého objemu nezměněných dat
  - Změny se následně přenesou do statistiky nad celou tabulkou
- Statistiky nad partition **nejsou využívány pro odhad kardinality**
  - Pokud potřebujete přesnější odhad kardinality, je třeba ručně vytvořit pro každou partition filtrované statistiky

Storage Tiering

- Za účelem archivace dat některé systémy kopírují starší data do archivních tabulek, které jsou uložené ve filegroups se soubory na pomalejších velkokapacitních discích
  - Takovou úpravu musí realizovat vývojáři aplikace, protože se musí počítat s odlišným přístupem k archivním datům
- Díky použití partitioningu jsme schopni realizovat uložení archivních dat do jiné filegroup ale v rámci jedné tabulky
  - Přístup je transparentní k aplikaci
  - Pokud není časová dimenze součástí clustered indexu, tak bude obtížné určit klíč partitioningu

Cílená komprese dat

- Microsoft SQL Server podporuje několik způsobů komprese dat, které se liší svou efektivitou ale i dopadem na výkon
  - Row Compression – Převádí datové typy s fixní délkou na variabilní délku a provádí kompresi uložených unicode textů
  - Page Compression – Row Compression + Prefix Compression + Dictionary Compression



Zdroj: https://docs.microsoft.com/en-us/sql/relational-databases/data-compression/page-compression-implementation

Cílená komprese dat

- Partitioning umožňuje nastavovat použitý způsob komprese a tedy omezit i výkonnostní dopady na úrovni jednotlivých partitions
  - Pro archivní data, kde je minimum modifikací dat, je možné aktivovat Page Compression, která je efektivní, ale snižuje výkon modifikací dat
  - Pro méně často modifikovaná data lze aplikovat Row Compression
  - Partition s aktivně modifikovanými daty by měla zůstat zcela bez komprese
- Míru komprese jednotlivých partitions lze postupně měnit s tím, jak aktivní partitions postupně přechází v archivní

Eskalace zámků

- Microsoft SQL Server používá pro řízení souběžného zpracování transakcí zámky, aby zajistil izolaci transakcí a podle zvolené izolační úrovně omezil problémy spojené se souběžným zpracováním
- Pro minimalizaci blokování, jako důsledku používaných zámků, jsou data zamykána po řádcích
- Pokud je zamknutých řádků v tabulce již mnoho, tak z důvodu úspory zdrojů dojde k uzamknutí celé tabulky
  - Eskalace zámků na úroveň tabulky
  - Dojde k omezení souběžného zpracování

Eskalace zámků

- Při použití partitioningu je možné povolit, aby docházelo k eskalaci zámků pouze na úroveň partition nikoliv celé tabulky
- Díky tomu velké množství zámků v rámci jedné partition neomezí souběžné zpracování v dalších partitions
- Výchozí chování tabulek je ale eskalace zámků na úroveň tabulky (level TABLE), je proto nutné změnit konfiguraci na úroveň AUTO:

```
ALTER TABLE dbo.Orders
SET (LOCK_ESCALATION = AUTO)
```

Efektivní vkládání nových a mazání starých záznamů (sliding window)

- Partition Switch umožňuje instantně vložit data do prázdné partition nebo naopak přenést data z partition do prázdné tabulky
- Je možné data postupně načíst do staging tabulky a následně instantně vložit do hlavní tabulky jako novou partition
- Mazání starých dat bývá velmi náročné na výkon, protože příkaz DELETE maže data po řádcích
  - Pokud chceme smazat celou starou partition, je možné ji přesunout do prázdné tabulky a tam použít velmi efektivní TRUNCATE TABLE
  - Lze dělat i TRUNCATE TABLE s uvedením partition (od Microsoft SQL Serveru 2016)

Efektivní vkládání nových a mazání starých záznamů (sliding window)

- Tento přístup vyžaduje, aby partitioning klíč byla časová dimenze
- Data budeme načítat a mazat s pomocí následujícího postupu:
  1. Pro vložení dat na konci intervalu s pomocí PARTITION SPLIT vytvoříme prázdnou partition, kam vložíme nová data
  2. Data načteme do staging tabulky a vložíme do připravené partition nebo můžeme zapisovat průběžně přímo do nové partition
  3. Mazaná data s pomocí PARTITION SWITCH přesuneme do prázdné staging tabulky, kde je s pomocí TRUNCATE TABLE smažeme
  4. Prázdnou partition na začátku odstraníme s pomocí PARTITION MERGE

Přetížení stránek v indexu

- SQL Server musí i na úrovni datových stránek v operační paměti řídit souběžný přístup k datovým stránkám pomocí **latches**
- Pokud veliké množství procesů chce zapisovat do téže stránky, je nutné přístup serializovat a stránku může modifikovat pouze jeden proces současně
- Častý problém při zápisu nových řádků do indexů v rostoucí posloupnosti klíče
  - Minimální fragmentace indexu při zápisu nových hodnot, ale všechny zápisy směřují do poslední stránky indexu

Přetížení stránek v indexu

- Pomocí partitioningu a volby vhodného partition klíče můžeme tuto zátěž rozložit mezi více B+ stromů jednotlivých partitions a jejich poslední stránky, kam se bude zapisovat
- Partition klíč bude třeba vybrat tak, aby docházelo přirozeně k distribuci vkládaných řádků mezi více partitions
  - Sloupec reprezentující čas není v tomto případně vhodným kandidátem
  - Např. u zápisu senzorových dat by bylo možné zvolit jako partition klíč identifikaci senzoru
  - Je třeba pamatovat na to, že by se partition klíč měl pak vyskytovat v dotazech, protože není efektivní hledat řádek ve všech partitions

Dotazy

Chcete si na toto téma společně povídat 5 dnů? Přijďte na GOC 631

RNDr. David Gešvindr, Ph.D.

MVP: Data Platform | MCSE: Data Platform | MCT

david@wug.cz

@gesvindr