

Optimalizace SQL dotazů

RNDr. David Gešvindr

MVP: Data Platform | MCSE: Data Platform | MCSD: Windows Store | MCT

david@wug.cz

 @gesvindr

Motivace

- Rostoucí výkon HW vede k mylné představě, že dotazy lze zpracovat „hrubou výpočetní silou“ bez nutnosti optimalizovat
- Přínosy optimalizace dotazů
 - Zvýšení výkonu aplikace – zvýšení propustnosti, zkrácení doby čekání na dokončení operace
 - Snížení nákladů na provoz – efektivnější využití HW (při stejné zátěži HW je obslouženo více operací)
- I zdánlivě nepatrná změna ve zpracování dotazu může způsobit řádové snížení délky trvání dotazů s minimálními náklady

Osnova

1. Aplikace jede pomalu, co s tím?
2. Jak nám pomohou indexy
3. Jak systematicky přistupovat k optimalizaci
4. Další zabijáci výkonu dotazů

Osnova

- 1. Aplikace jede pomalu, co s tím?**
2. Jak nám pomohou indexy
3. Jak systematicky přistupovat k optimalizaci
4. Další zabijáci výkonu dotazů

Metody monitorování T-SQL dotazů a serveru

- SQL Server Profiler
- SQL Trace
- Extended Events

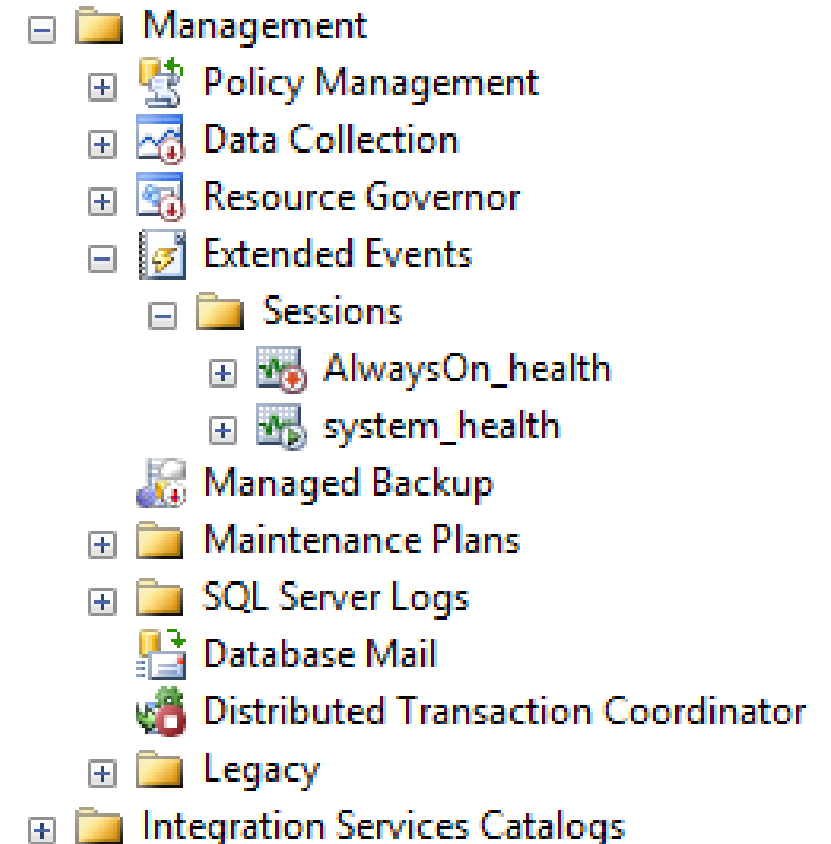
- Dynamic Management Views and Functions
- SQL Server Management Studio
 - Activity Monitor
 - Reports
- Performance Monitor
- Data Collector
- Query Store

SQL Server Profiler

- Grafický nástroj pro **trasování**
 - Zachytávání událostí, které se stanou v SQL Serveru
- Kdy se používá
 - Identifikace náročných dotazů
 - Zachycení zátěže pro SQL Server Tuning Advisor
 - Zachycení zátěže pro „replay“
 - Sběr informací o dalších událostech SQL Serveru (deadlock)
 - Sledování bezpečnostních událostí
- **V SQL Serveru 2016 je deprecated, používejte Extended Events**

Extended Events

- Technologie, která nahrazuje SQL Trace a dále rozšiřuje jeho možnosti
 - SQL Trace je označen jako deprecated
- Je k dispozici od SQL Serveru 2008
- Ale v Management Studiu je podpora až od **SQL Serveru 2012**
- Způsobuje minimální zátěž pro SQL Server
- Konfigurace s pomocí přehledného T-SQL kódu



Osnova

1. Aplikace jede pomalu, co s tím?
- 2. Jak nám pomohou indexy**
3. Jak systematicky přistupovat k optimalizaci
4. Další zabijáci výkonu dotazů

Jak SQL Server ukládá data

- SQL Serveru ukládá data do 8 KB datových stránek
- Typy datových stránek:
 - Global allocation map and shared global allocation map
 - Page free space
 - Differential changed map
 - Bulk changed map
 - **Index allocation map**
 - **Index**
 - **Data**
 - Text or image
- 8 po sobě jdoucích datových stránek tvoří **extent**
 - Mixed extents
 - Uniform extents

Jak zjistit složitost dotazu

- **Exekuční plán** popisuje, jaké fyzické operace SQL Server musel realizovat pro načtení dat
 - Je vyčíslena cena dotazu
 - **Pozor – Tato cena je vypočtena na základě odhadovaného exekučního plánu a nemusí odpovídat skutečnosti!**
- **SET STATISTICS IO ON**
 - SQL Server ukáže počet I/O operací spojených s exekucí dotazu
- **SET STATISTICS TIME ON**
 - SQL Server ukáže pro každý příkaz v dávce dobu kompilace a exekuce

Jak jsou uložena data v tabulce

Heap

- Nemá specifické řazení po sobě jdoucích stránek
 - IAM udržuje pouze seznam stránek, které tvoří tabulku
- Nemá specifické řazení řádků ve stránkách
- Výhodou je, že nové nebo změněné řádky mohou být umístěny kamkoliv do stránek haldy
- Nevýhodou je, že bez neclusterovaného indexu musíme pro najetí hodnoty projít všechny datové stránky tabulky

Clustered Index

- Datové stránky tabulky jsou logicky seřazeny
- Řádky ve stránkách jsou seřazeny dle klíče indexu
- Možnost vytvoření pouze jediného clusterovaného indexu
 - Nechceme 100% duplicitu uložených dat
- B+ strom který v listech obsahuje celé řádky

Osnova

1. Aplikace jede pomalu, co s tím?
2. Jak nám pomohou indexy
- 3. Jak systematicky přistupovat k optimalizaci**
4. Další zabijáci výkonu dotazů

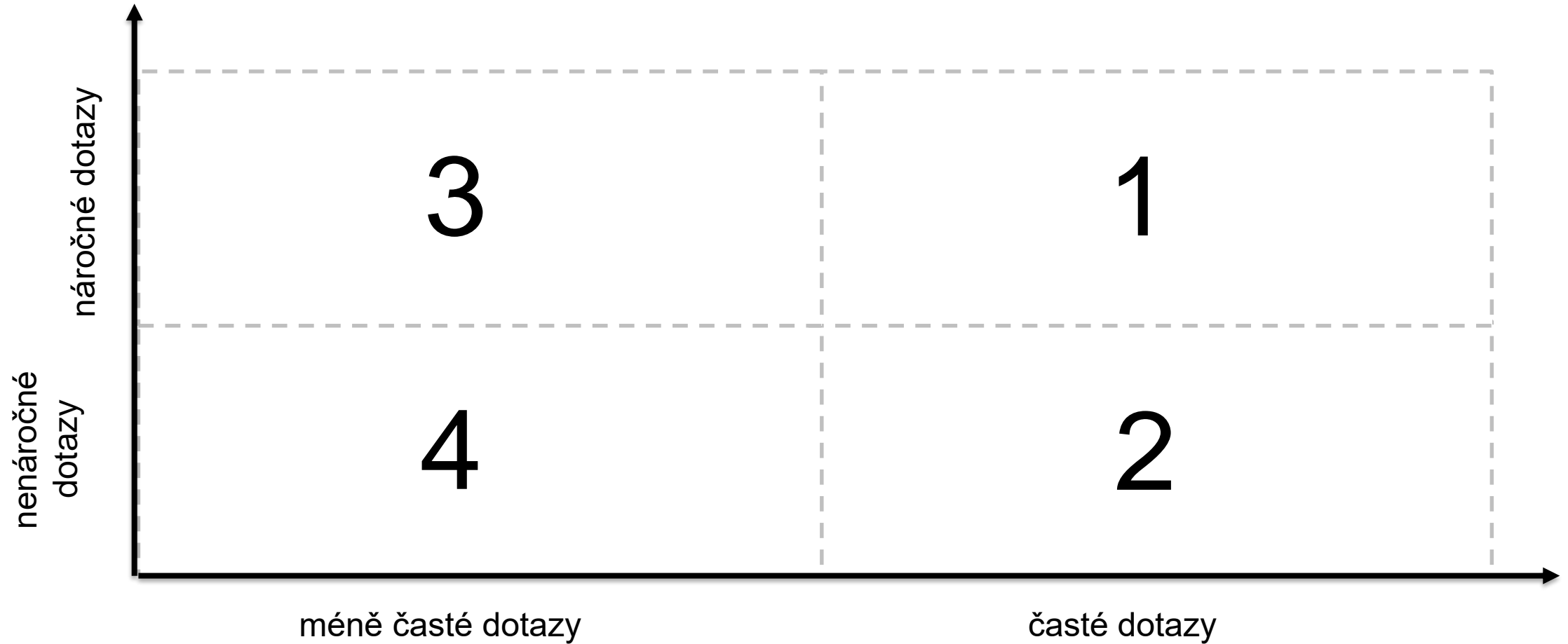
Cíl optimalizace

- Ne všechny operace mohou být efektivně optimalizovány
- Vždy je třeba uvážit poměr mezi **náklady na optimalizaci** a **jejími přínosy**
- **Je třeba si stanovit předem reálné cíle optimalizace a tomu podřídit celý proces optimalizace**
 - Konkrétně popsat, jaké jsou požadavky na výsledky optimalizace
 - Např.: Načtení detailu objednávky musí trvat méně než 1s

Paretovo pravidlo

- Všeobecně známé **pravidlo 80/20**
- **80% důsledků pramení z 20% příčin**
- **Paretovo pravidlo je aplikovatelné i na proces optimalizace dotazů**
- Optimalizací **20% nejnáročnějších dotazů** zvýšíme výkon aplikace až o **80%**

Cíl optimalizace



Query Store

EXPRESS

- Nový způsob monitorování náročných dotazů dostupný od SQL Serveru 2016
- Sbírá data o:
 - Exekuci jednotlivých dotazů
 - Používaných variantách exekučních plánů
- Použití:
 - Identifikace náročných často spouštěných dotazů
 - Identifikace dotazů, kde se zhoršila exekuce
 - Porovnání dopadů optimalizace

Aktivace Query Store

- Aktivace a konfigurace Query Store se provádí ve vlastnostech databáze

Database Properties - AdventureWorks

Select a page

- General
- Files
- Filegroups
- Options
- Change Tracking
- Permissions
- Extended Properties
- Mirroring
- Transaction Log Shipping
- Query Store

Script Help

General

Operation Mode (Actual)	Off
Operation Mode (Requested)	Read Write

Monitoring

Data Flush Interval (Minutes)	15
Statistics Collection Interval	1 Hour

Query Store Retention

Max Size (MB)	100
Query Store Capture Mode	All

Operation Mode (Requested)
The requested query store operation mode.

Current Disk Usage

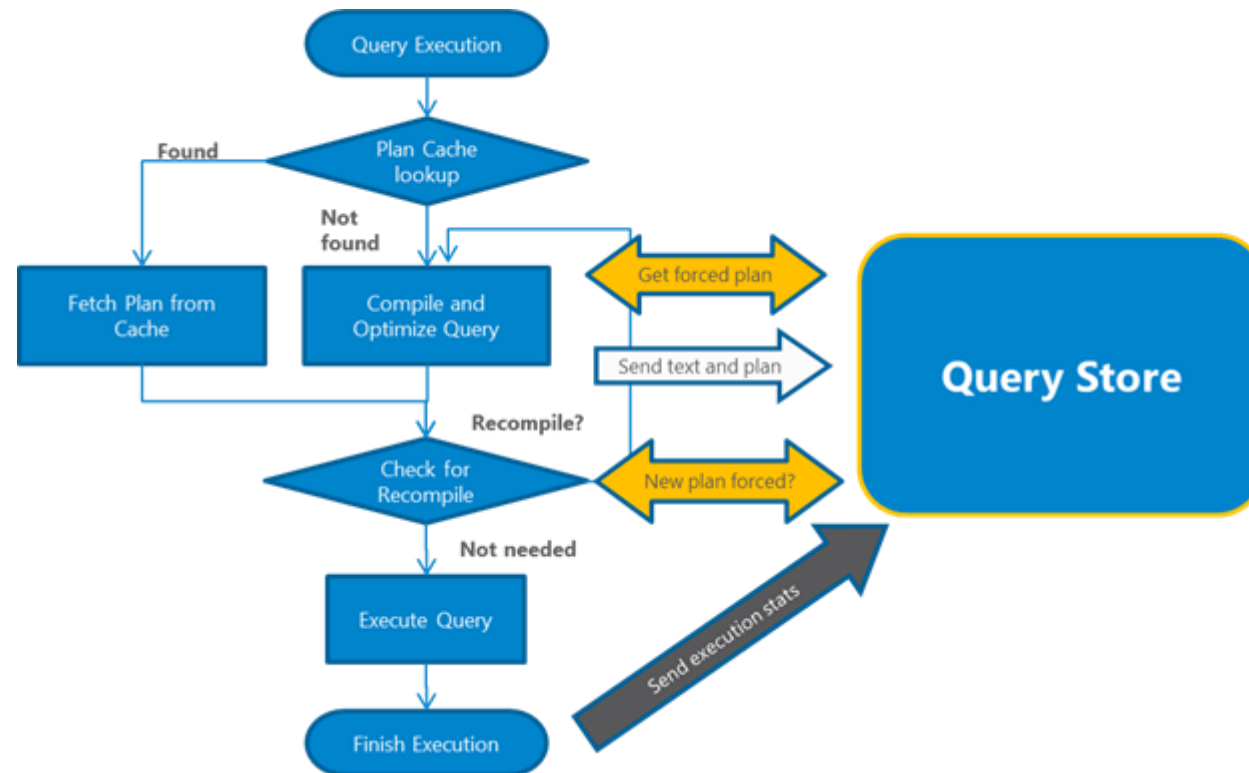
AdventureWorks 223.3 MB	Query Store Available 100.0 MB
Query Store Used 0.0 MB	Query Store Used 0.0 MB

Purge Query Data

OK Cancel

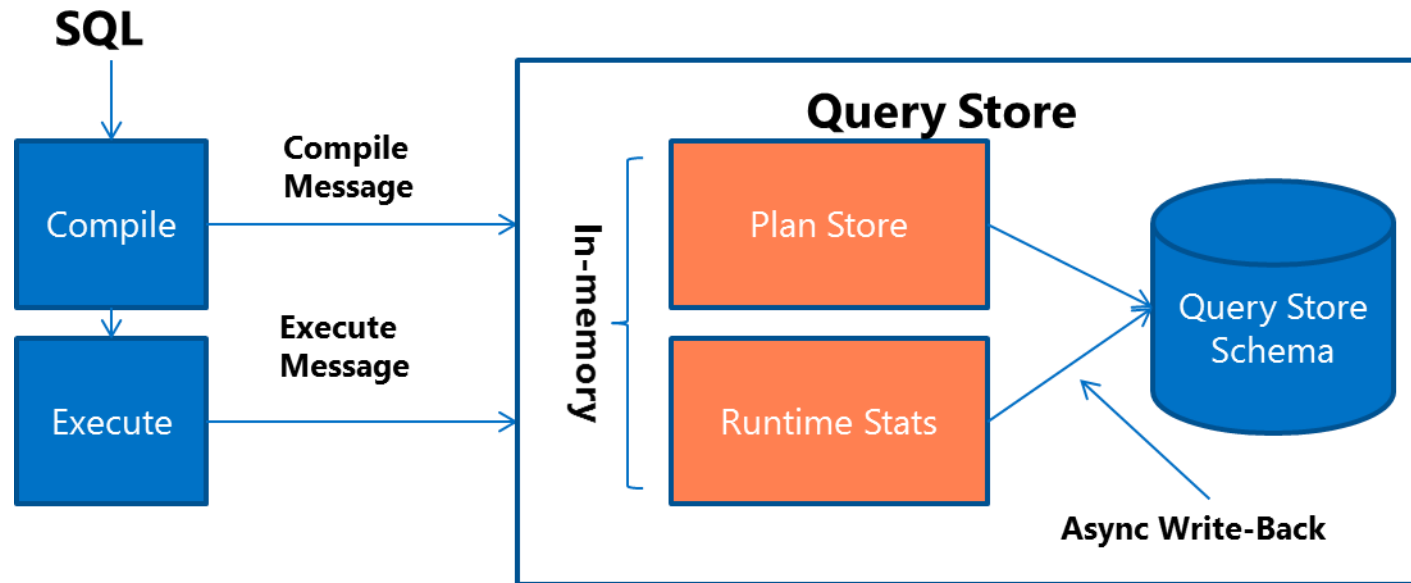
Sběr dat pro Query Store

- Query Store po povolení automaticky sbírá data o kompilaci exekučního plánu včetně statistik o jeho exekuci



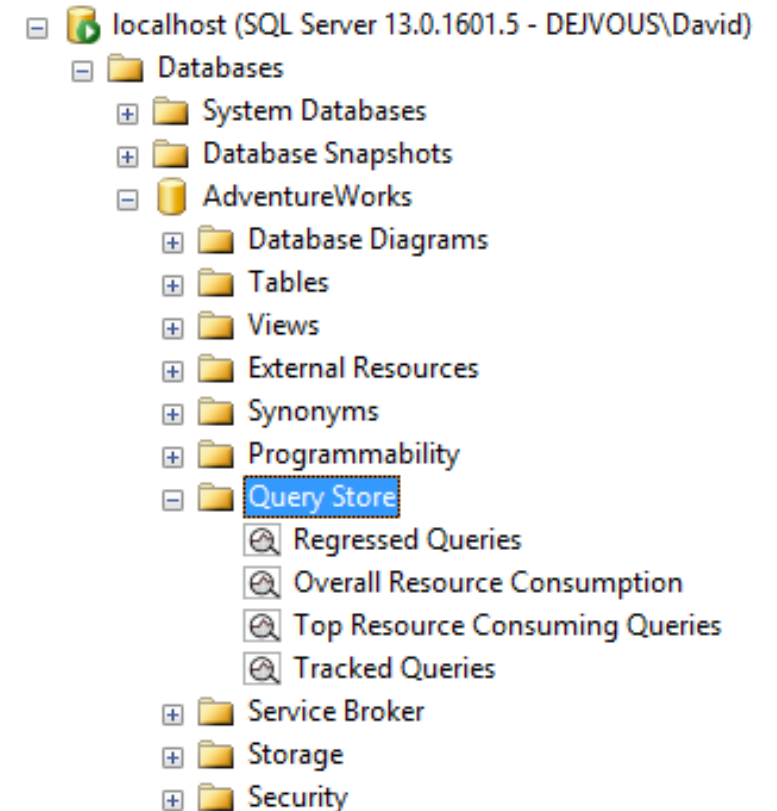
Výkonnostní dopad Query Store

- Query Store by mělo mít minimální dopad na výkon SQL Serveru
- Query Store je navrženo tak, aby data cachovalo v operační paměti a dávkově zapisovalo na disk



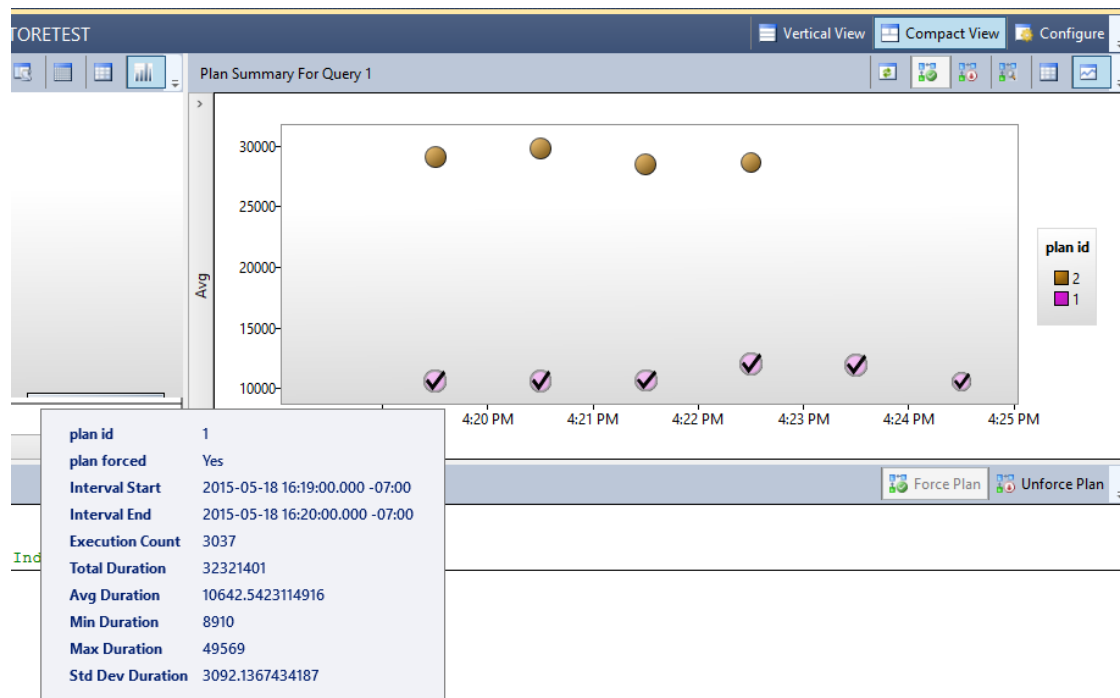
Vizualizace dat

- Query Store přidává do SQL Server Management Studio 2016 novou sekci
- Hlavní pohledy na data:
 - Regressed Queries
 - Overall Resource Consumption
 - Top Resource Consuming Queries
 - Tracked Queries



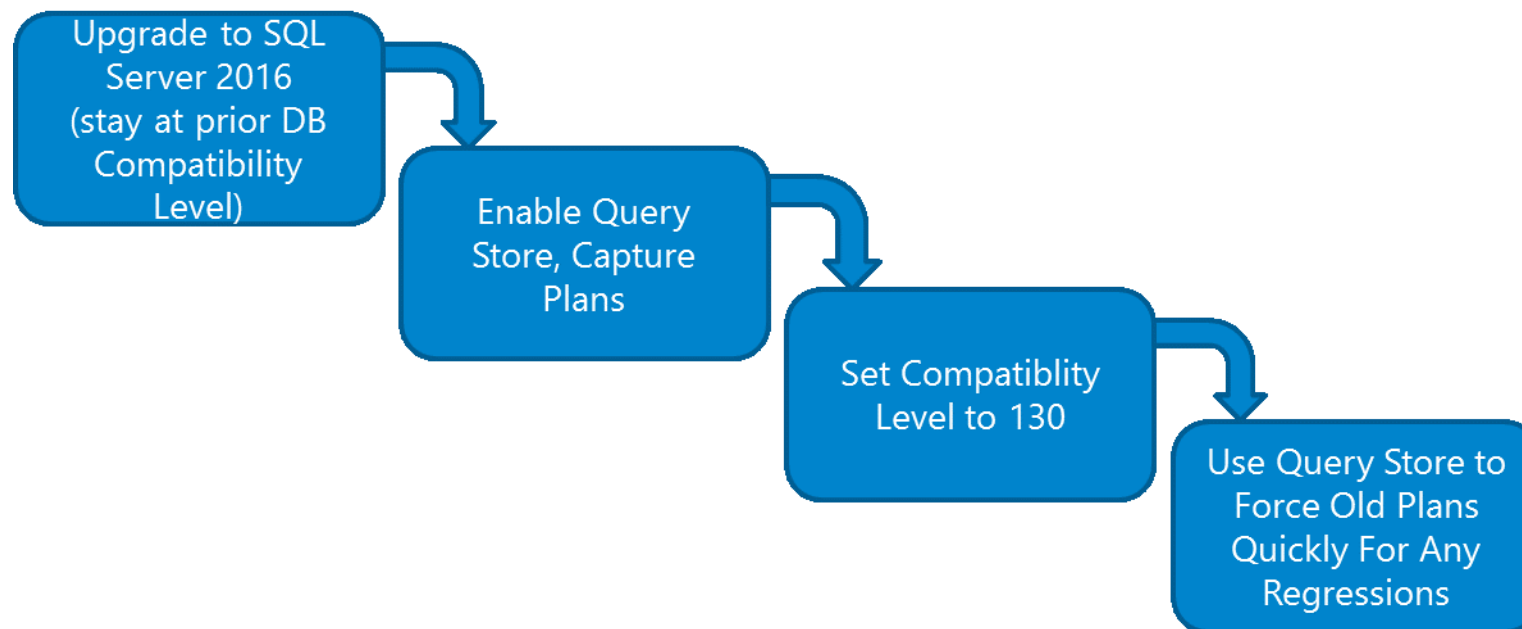
Použití: Identifikace alternativních plánů

- V současnosti je problém identifikovat různé varianty exekučních plánů stejného dotazu a porovnat jejich efektivitu (plan choice change regression)



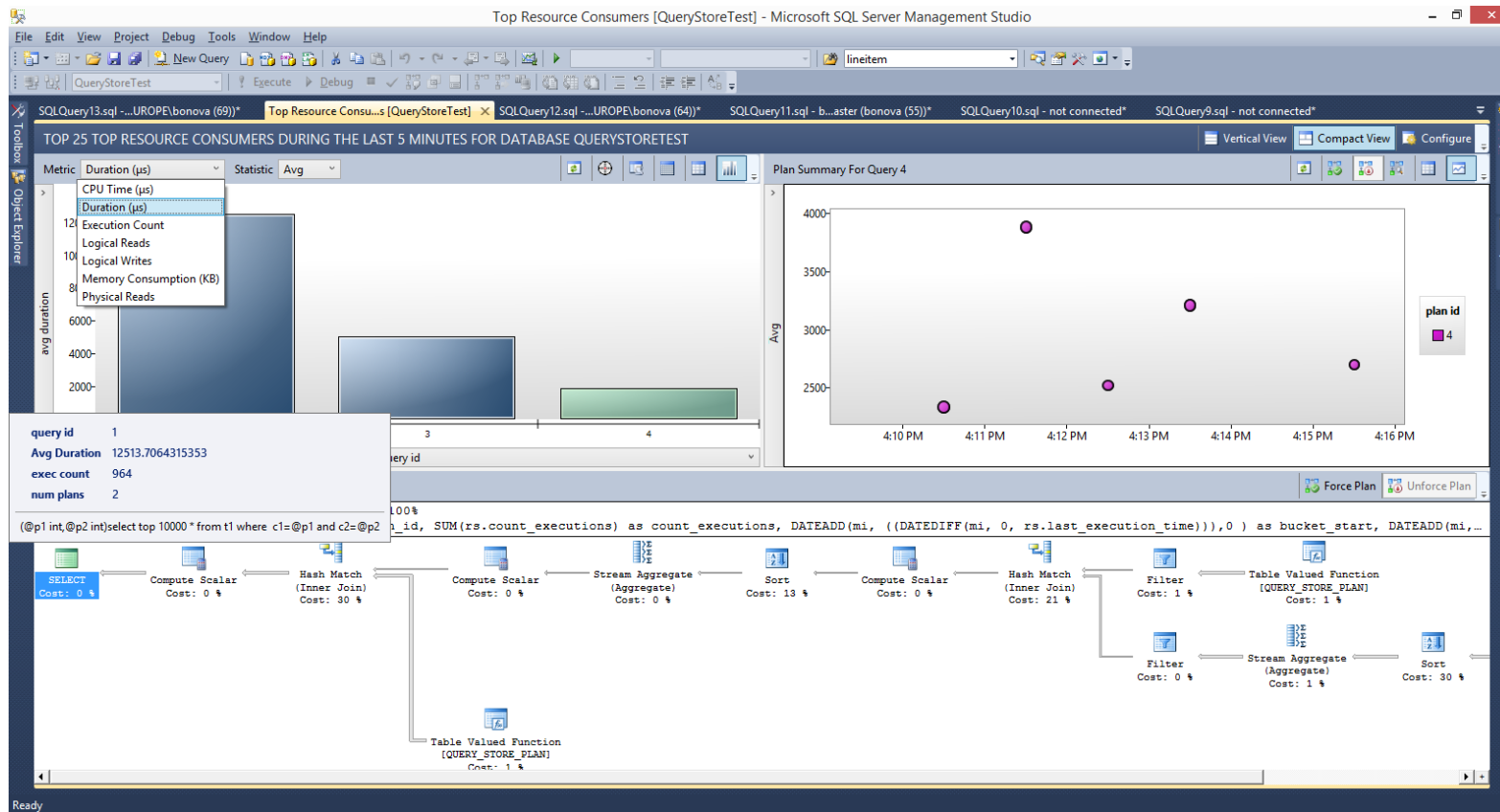
Použití: Spolehlivá migrace na novou verzi

- Při upgradu na novou verzi SQL Serveru může docházet ke generování odlišných exekučních plánů
- Query Store tyto situace umí detekovat a opravit



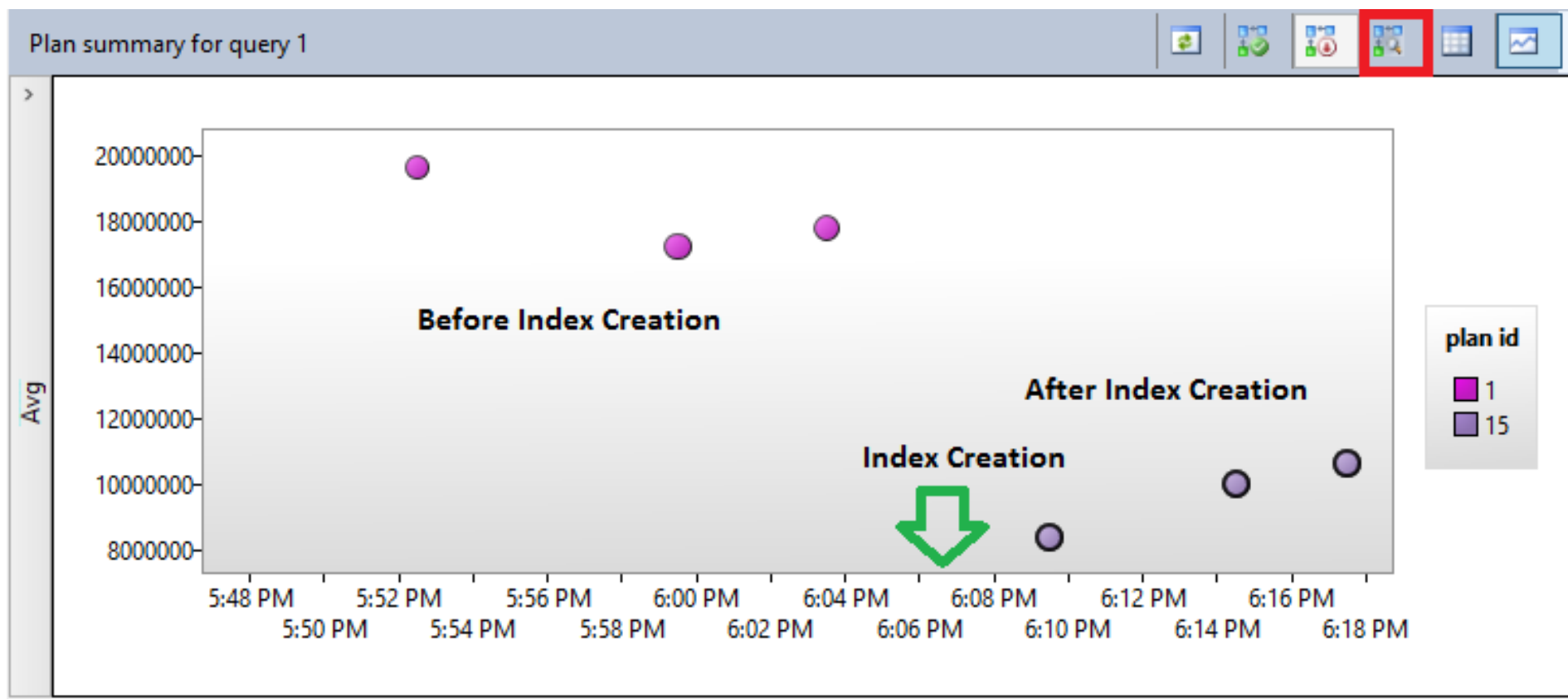
Použití: Identifikace nejnáročnějších dotazů

- Díky sběru informací o exekuci dotazů umí Query Store identifikovat nejnáročnější dotazy podle různých kritérií



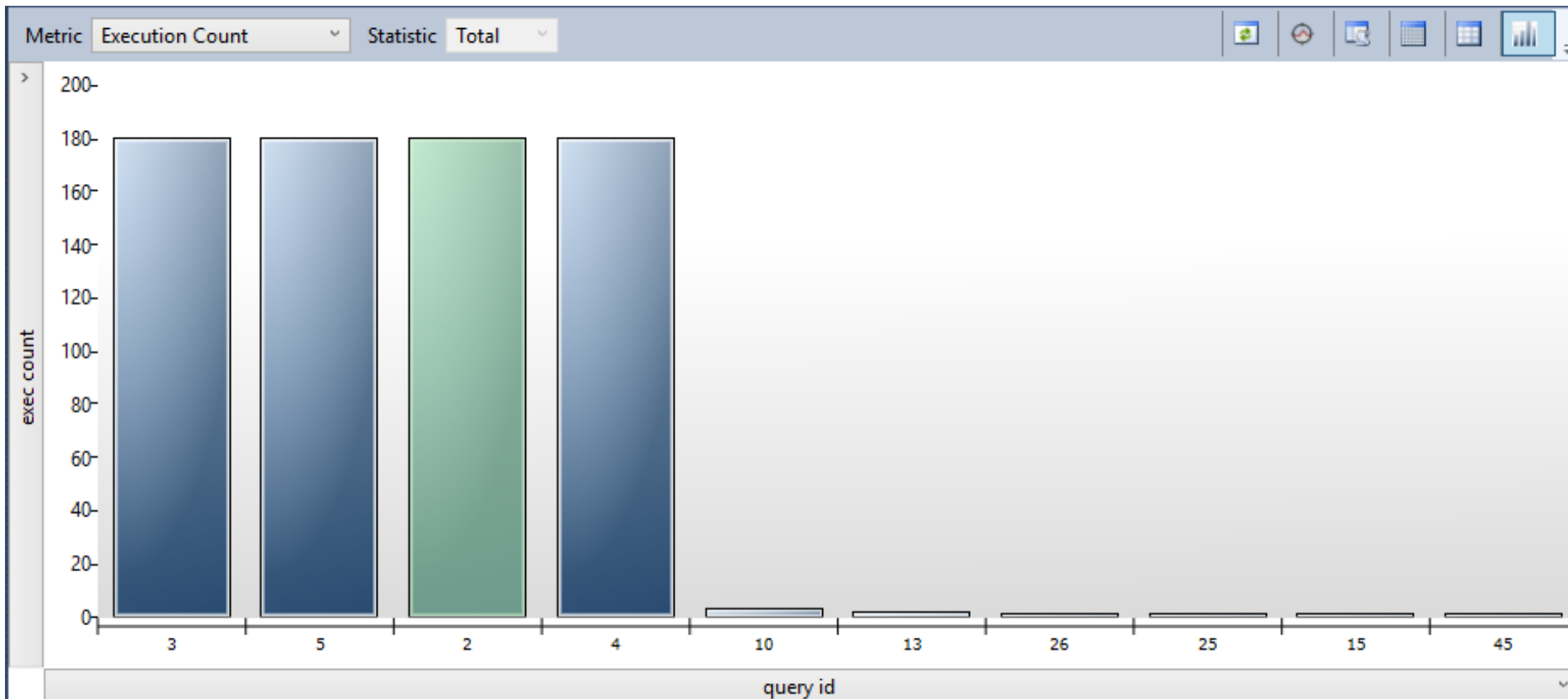
Použití: A/B testování

- Po provedení optimalizace je možné měřit a analyzovat její dopad



Použití: Optimalizace ad-hoc dotazů

- Po rekonfiguraci SQL Serveru jsou dotazy správně parametrizovány



Osnova

1. Aplikace jede pomalu, co s tím?
2. Jak nám pomohou indexy
3. Jak systematicky přistupovat k optimalizaci
4. **Další zabijáci výkonu dotazů**

Nevhodný návrh dotazu

- Efektivní dotaz načítá z databáze jen nutné minimum informací, které dostačuje aplikaci pro daný účel
 - Např.: Pro vypsání seznamu objednávek nemusím načítat 50 dalších sloupců v tabulce objednávek když nejsou zobrazeny v UI
- Je důležité:
 - Vracet jen sloupce, co skutečně využijeme (pozor na `SELECT * FROM tabulka`)
 - Filtrovat a stránkovat záznamy na serveru
 - Zbytečně neřadit záznamy, pokud to opravdu nepotřebujeme

Nevhodný návrh dotazu

- V některých případech složitost dotazu překročí optimalizační schopnosti SQL Serveru a ten sestaví neefektivní exekuční plán
- Pokud nepomohou jiné optimalizační techniky, může být řešením dotaz rozdělit na více menších dotazů
- Pozor na režii spojenou s uložením dat do dočasných tabulek

Non-set based operace

- Dotazy v T-SQL popisují jaká data chceme získat, ale nepopisují jak je získat
- Efektivní způsob načtení dat je pak sestaven v rámci optimalizace dotazu
- SQL Server je optimalizovaný na množinové operace
 - Pokud budeme pracovat s daty řádek po řádku, bude to mít významný dopad na výkon dotazu
 - Ne vždy se jde vyhnout kurzorům nebo cyklům, ale měli bychom se o to snažit

Nadměrné uzamykání a deadlock

- SQL Server pro řízení souběžného přístupu používá mechanismus zámků
- Kvůli použití zámků dochází k blokování
 - To není problém, pokud netrvá dlouho
- Pokud zpracováváme rozsáhlé transakce a zvolíme nevhodnou úroveň izolace, vzniká velké množství zámků, které blokují další operace až do konce transakce
- Pokud k datům přistupujeme v nevhodném pořadí, může dojít k vzájemnému uváznutí - deadlock

Špatný návrh databáze

- Tabulky v relačních databázích navrhujeme s využitím principů normalizace
- Výhody normalizované databáze:
 - Odstranění duplicity dat a anomálií při změnách
- Další projevy špatného návrhu databáze:
 - Potřeba nadměrně spojovat tabulky
 - Nevhodné použití indexů
 - Příliš rozsáhlé zámky

Osnova

1. Aplikace jede pomalu, co s tím?
2. Jak nám pomohou indexy
3. Jak systematicky přistupovat k optimalizaci
4. Další zabijáci výkonu dotazů

Dotazy

RNDr. David Gešvindr

MVP: Data Platform | MCSE: Data Platform | MCSD: Windows Store | MCT

david@wug.cz

 @gesvindr