

# Novinky v Microsoft SQL Serveru 2017

**RNDr. David Gešvindr**

MVP: Data Platform | MCSE: Data Platform | MCT

[david@wug.cz](mailto:david@wug.cz)

 @gesvindr

# Osnova

1. Modern Servicing Model
2. SQL Graph
3. Adaptive Query Processing
4. Automatic Database Tuning
5. Wait Stats v Query Store
6. Temporal Tables Retention Policy
7. Vylepšení In-Memory OLTP
8. Nové užitečné DMV
9. SQL Server on Linux
10. Clusterless AlwaysOn Availability Groups

# Modern Servicing Model

- Microsoft již nebude vydávat **Service Pack (SP)** od SQL Serveru 2017
  - Nevztahuje se na dřívější verze, ty obdrží Service Pack po dobu hlavní podpory
- Hlavní zdroj aktualizací budou **Cummulative Updates (CU)**
  - Budou obsahovat opravy chyb, nové funkce a lokalizovaný obsah
  - První rok vydávány po 1 měsíci, následné 4 roky po 3 měsících až do konce 5 leté podpory
  - Jednou za rok Microsoft vydá nové instalační médium včetně aktualizací
- Microsoft bude dle potřeby vydávat **General Distribution Releases (GDR)** opravující bezpečnostní chyby (aplikovatelné na RTM verzi)

# Servicing Lifecycle

- 0. – 5. rok – **Mainstream Support**
  - Vydávání Cumulative Updates, opravy chyb, drobné rozšiřování funkcionality
- 6. – 10. rok – **Extended Support**
  - Microsoft vydává pouze bezpečnostní aktualizace nebo opravy kritických chyb
- 11. – 15. rok - **Premium Assurance**
  - Placená podpora

## Kdy bude další verze SQL Serveru?

# SQL Graph

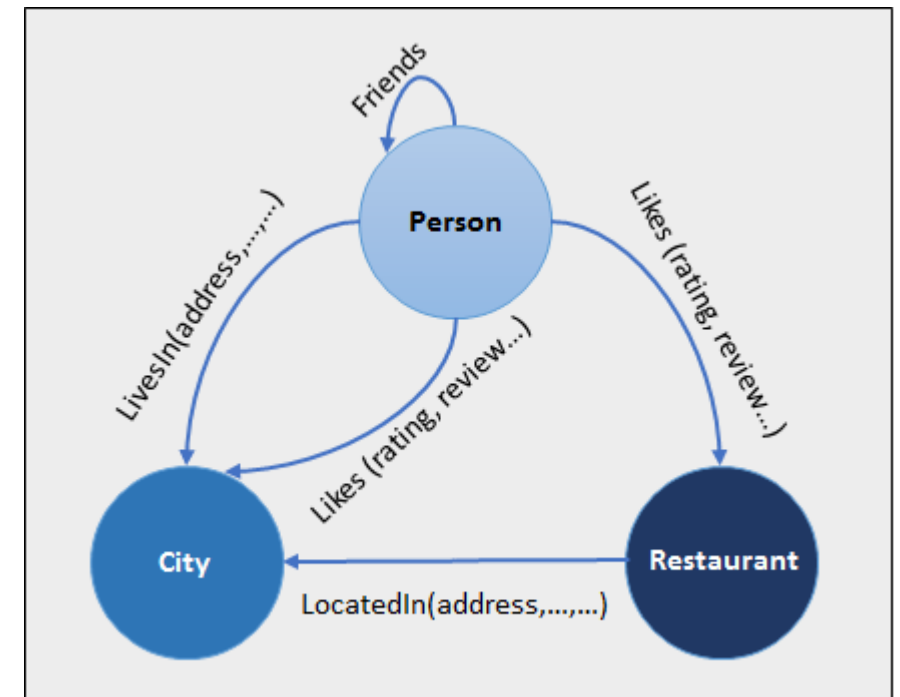
- SQL Server 2017 přidává podporu **uložení a dotazování grafových dat** přímo v relační databázi
- Grafovými daty rozumíme:
  - Uložení **uzlů**, které mají své atributy
  - Libovolné **propojení uzlů do grafu pomocí hran**, které mohou mít také své atributy

# Výhody grafové databáze

- Uzly reprezentují entity včetně jejich vlastností
- Hrany reprezentují vztahy mezi entitami
  - Grafová databáze umí jednoduše vyjádřit vztahy m:n
  - Relační databáze umí uložit vztahy 1:n

- Ukázková databáze:

<https://docs.microsoft.com/en-us/sql/relational-databases/graphs/sql-graph-sample>



# Dotazování grafu s pomocí klauzule MATCH

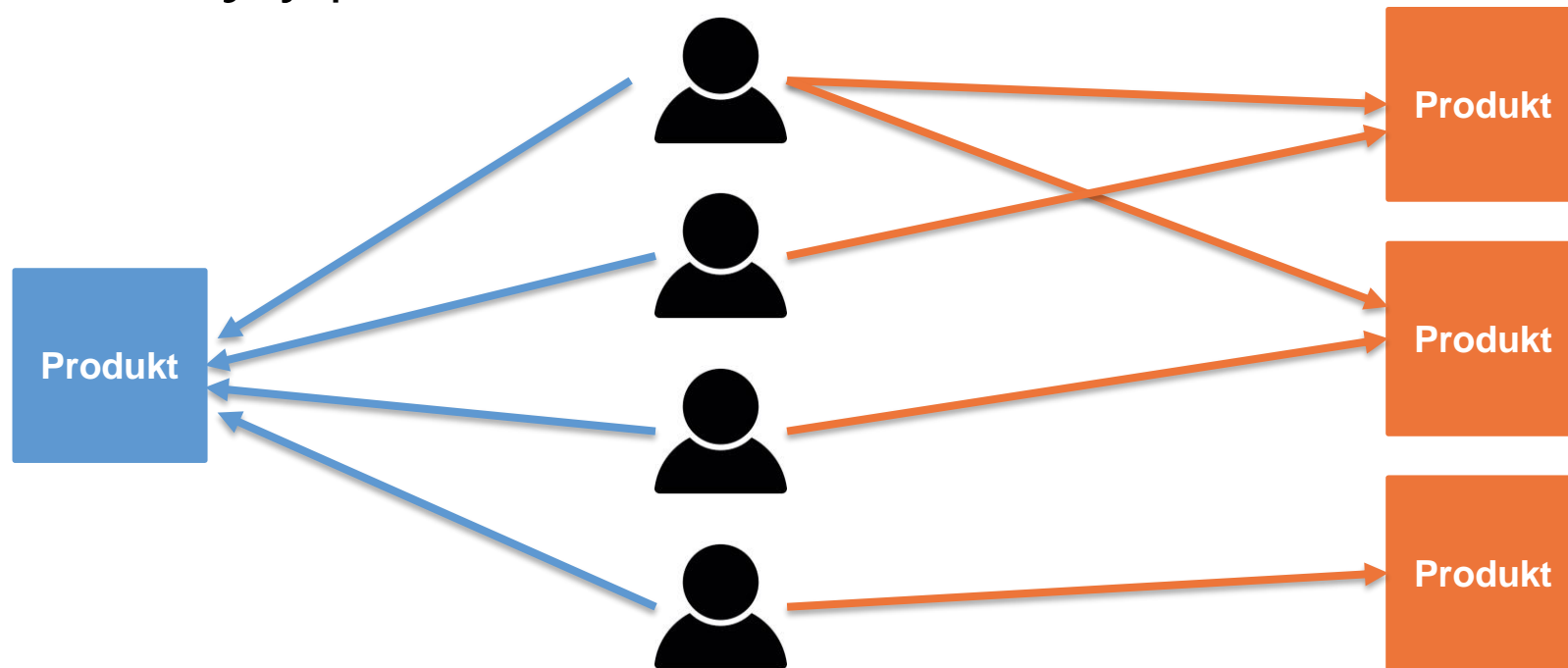
- Pro dotazování v grafu slouží nový predikát **MATCH**

```
SELECT person2.[Name]  
FROM Person person1, friendOf, Person person2  
WHERE MATCH(person1-(friendOf)->person2)  
AND person1.[Name] = 'John';
```

# Zajímavé scénáře použití

## ▪ Doporučené produkty (frequently bought together)

- Doporučené produkty podle toho, co si koupili další uživatelé, kteří si koupili ten stejný produkt





# Chybějící funkcionalita

- **Podpora více grafů**
  - Uzly a hrany nyní tvoří jeden graf
  - Omezení, mezi kterými uzly mohou vést které hrany
- **Tranzitivní uzávěr (transitive closure)**
  - Musíme používat rekurzivní CTE na hledání spojení mezi uzly o neznámé délce
- **Polymorphism**
  - Nemůžeme dotázat všechny uzly (nehledě na typu) spojené s jiným uzlem
  - Obchází se s pomocí UNION a rozdělením na dotazy pro každý typ uzlu
- **Pokročilé algoritmy** (např. nejkratší cesta grafem)

# Další omezení

- Dočasná tabulka nemůže být uzel nebo hrana
- Tabulky a hrany nelze verzovat s pomocí **system versioned tables**
- Uzly a hrany nemohou být memory optimized tabulky
- U hrany nelze aktualizovat **\$from\_id** a **\$to\_id** sloupce – hranu je nutné smazat a založit znovu mezi jinými uzly

# Adaptive Query Processing

- Exekuční plán je sestavený na základě informací, které má optimalizátor k dispozici **před spuštěním dotazu**
- Možné problémy
  - Predikce kardinality – nepřesné statistiky, Multi-statement Table Valued Function
  - Parameter Sniffing

# Adaptive Query Processing

- Adaptive Query Processing vnáší do generování exekučních plánů **zpětnou vazbu ze současné nebo minulé exekuce dotazu**
- V SQL Serveru 2017 jsou podporovány 3 technologie:
  - Batch Mode Memory Grant Feedback
  - Batch Mode Adaptive Join
  - Interleaved Execution
- Pro aktivaci Adaptive Query Processingu je třeba přepnout compatibility level databáze na úroveň 140

# Batch Mode Memory Grant Feedback

- Dotaz pro své zpracování potřebuje přidělenou paměť
  - Náročné jsou operátory Sort a Hash Join
- Paměť se odhaduje na základě počtu řádků zpracovávaných jednotlivými operacemi v exekučním plánu
- Pokud dotaz obdrží paměti více, než potřebuje, snižujeme množství souběžně zpracovávaných dotazů
- Pokud paměti obdrží nedostatek, dojde k tempdb spills, které znatelně zpomalí exekuci dotazu

# Batch Mode Memory Grant Feedback

- SQL Server si u exekučního plánu v cache exekučních plánů nově ukládá zpětnou vazbu z minulé exekuce – kolik paměti bylo skutečně třeba
  - Pokud bylo třeba aspoň 2x méně paměti, opraví se plán (nad 1 MB)
  - Pokud se aktivuje disk spill, dojde k přepočtu potřebné paměti v plánu
- Tuto informaci zohledňuje při následné exekuci dotazu
- Pokud by hodnota prudce oscilovala (nerovnoměrné rozložení hodnot), tato funkce se automaticky zablokuje

# Batch Mode Adaptive Join

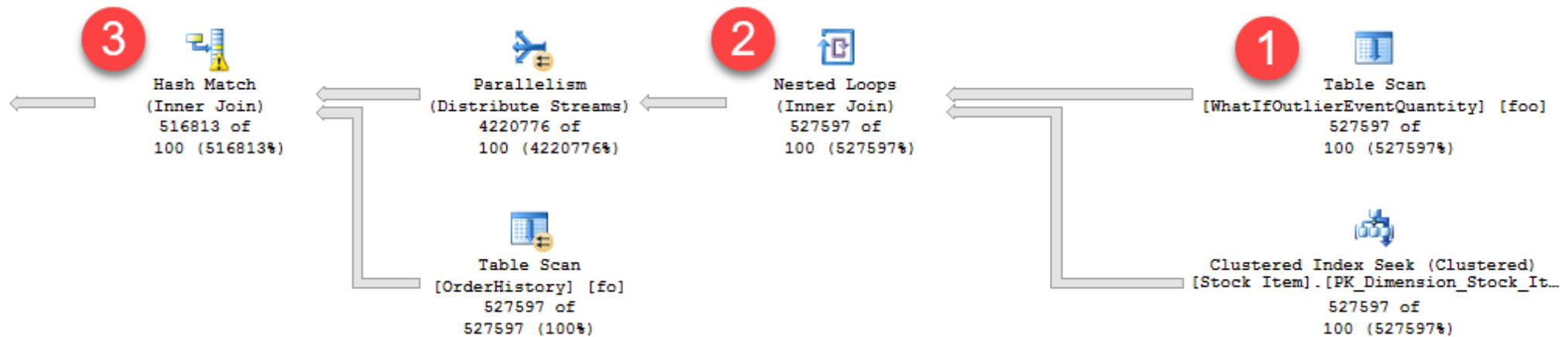
- Na základě odhadovaného počtu řádků se SQL Server rozhoduje v exekučním plánu mezi použitím algoritmu **nested loop** a **hash join**
- **Pokud se odhad liší od skutečnosti, může to mít fatální následky na výkon dotazu**
  - Problém i s parameter sniffingem, kdy se plán sestaví podle vstupních parametrů u první exekuce procedury





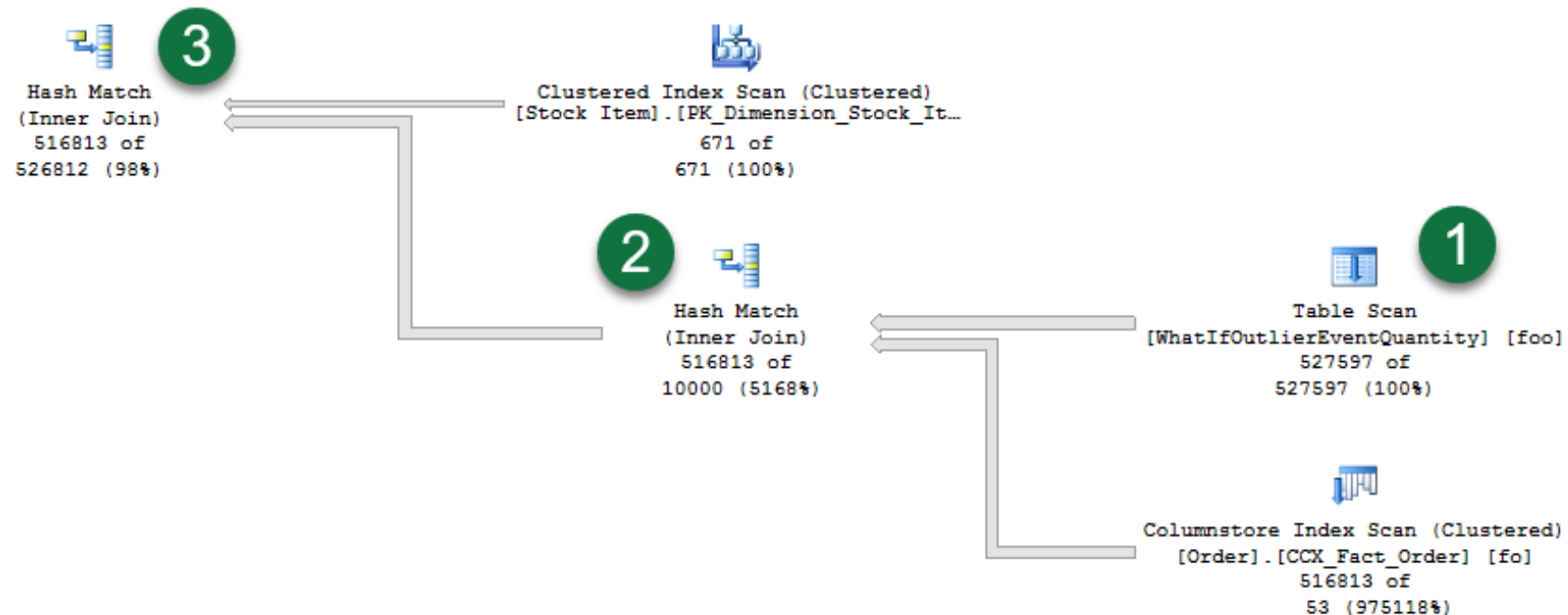
# Interleaved Execution

- Multi-statement Table Valued Functions mají kardinalitu v exekučním plánu pevně stanovenou na 100 (od SQL Serveru 2014, dříve 1)

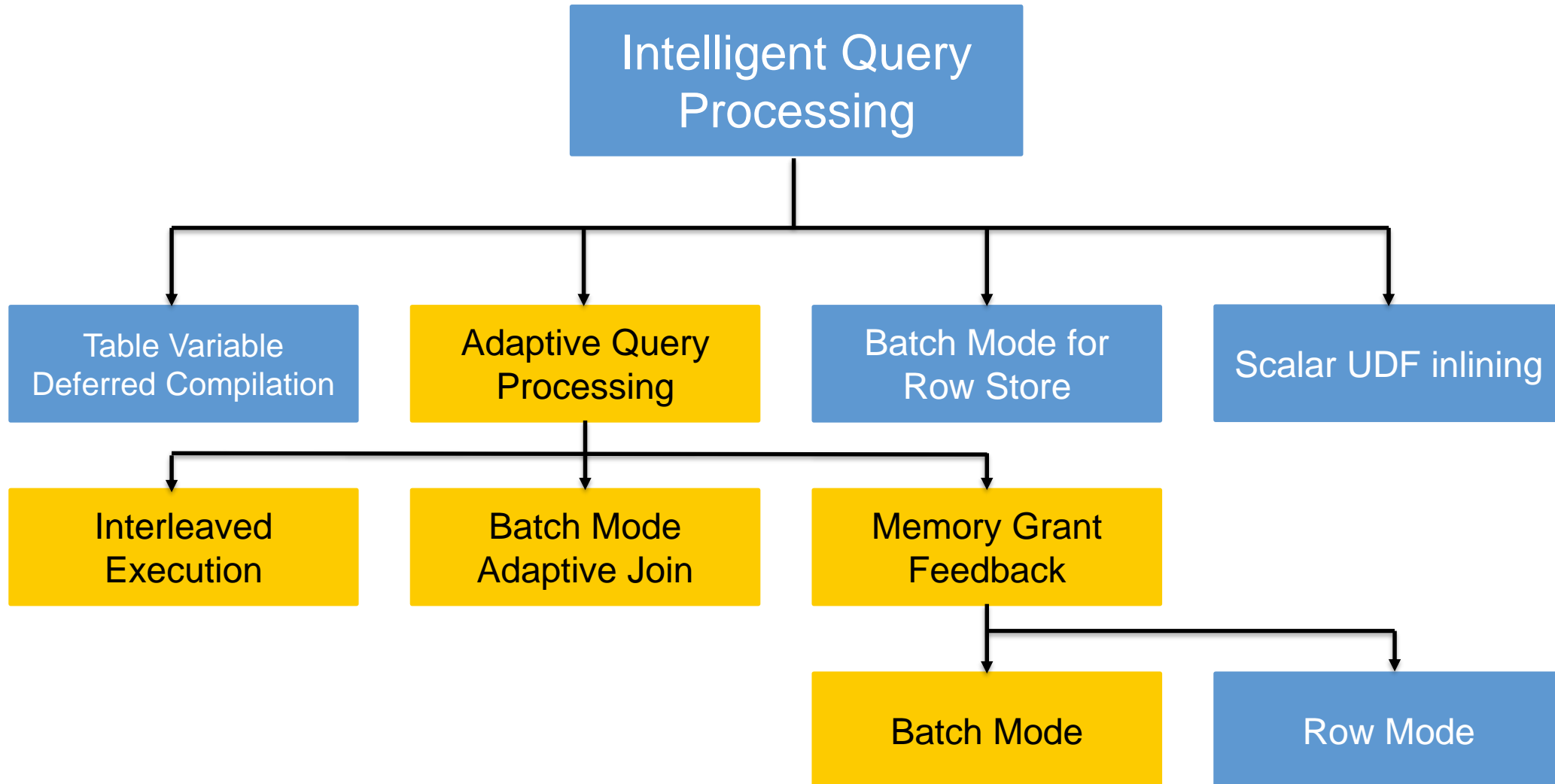


# Interleaved Execution

- Interleaved Execution umí přerušit proces optimalizace dotazu, vyhodnotit neznámý podstrom a pro pokračování optimalizace použít skutečný počet řádků



# Co bude dál?



# Automatic Database Tuning

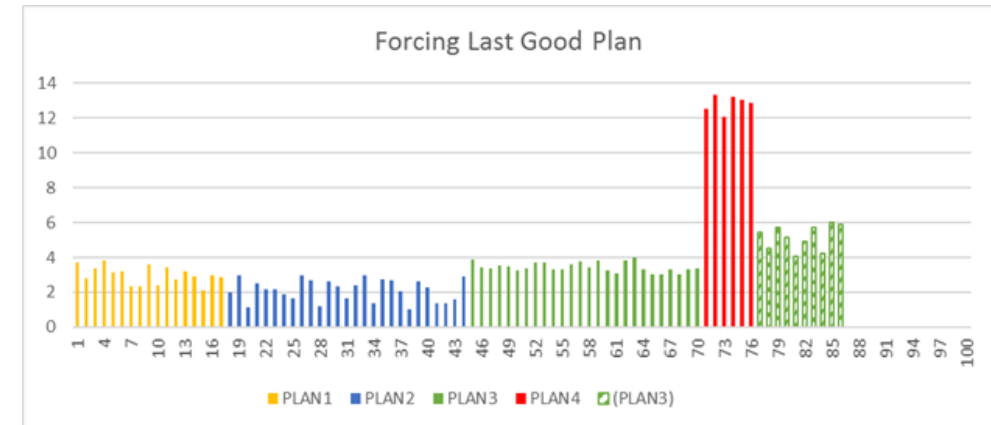
- Azure SQL Database podporuje **Automatic index management**
  - Přidává indexy, které chybí
  - Odstraňuje zbytečné indexy
- SQL Server 2017 podporuje **Automatic plan correction**
  - Detekuje **SQL plan choice regression** a umí ji automaticky napravit

# Automatic Plan Correction

- Pokud dojde ke změně exekučního plánu dotazu a nový plán vykazuje horší výkon, je výhodné využít předchozí plán
- Toto je možné detekovat přes Query Store
- Pravidelná manuální detekce, vynucování přechozího plánu a monitoring jeho výkonu je časově náročná činnost
  - SQL Server 2017 nabízí **Automatic Plan Correction**

# Automatic Plan Correction

- Doporučení jsou dostupná v **sys.dm\_db\_tuning\_recommendations**
  - Typ doporučené akce (FORCE\_LAST\_GOOD\_PLAN)
  - Popis problému
  - Skóre



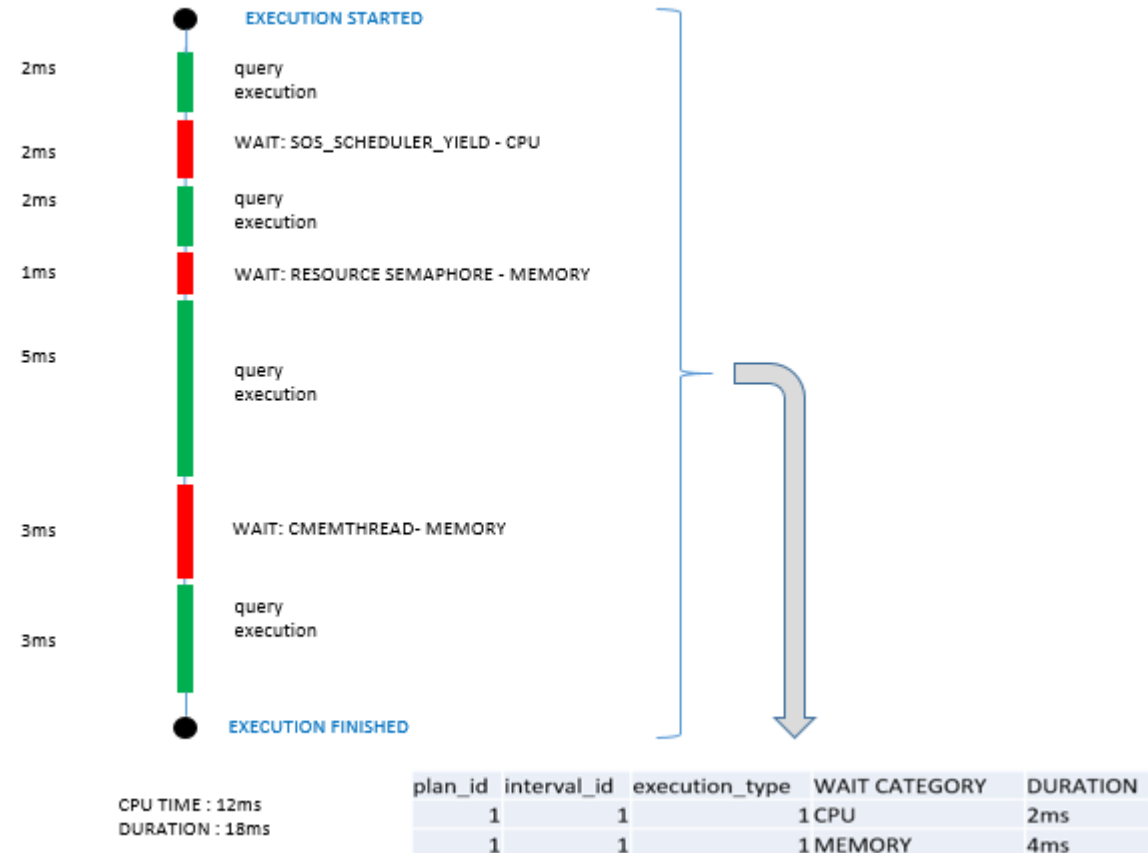
```
ALTER DATABASE AdventureWorks2014_  
SET AUTOMATIC_TUNING (FORCE_LAST_GOOD_PLAN = ON);
```

- Vynucený plán je aktivně monitorován a v případě, že začne být horší než ten původní je vynucení zrušeno

# Wait Stats v Query Store

- Máte problémy s dlouho běžícími dotazy a nevíte, na co čekají?
- Query Store nyní ukládá informace o tom „na co dotaz čeká“
- Agregováno dle kategorie čekání
- Data v [sys.query\\_store\\_wait\\_stats](#)

Single query execution timeline



# Temporal Tables Retention Policy

- SQL Server přináší novou funkcionalitu Temporal Tables
  - Tabulka automaticky ukládá všechny změněné nebo smazané verze záznamů
  - Doba uložení byla nastavena na nekonečno
- SQL Server 2017 umožňuje nastavit nad těmito historickými daty **retenční politiku**

```
ALTER TABLE dbo.Speakers  
SET (SYSTEM_VERSIONING = ON (HISTORY_RETENTION_PERIOD = 9 MONTHS));
```

- Mazání probíhá efektivně (B-tree Index, Columnstore Index)



# In-memory OLTP

- SQL Server 2014 přidává podporu pro **Memory Optimized Tabulky** a **nativně kompilované uložené procedury**
- Jedná se o nový engine pro uložení a zpracování dat přímo v operační paměti
  - Nevyužívá zámky, ale multi-version optimistic concurrency
  - Je zaručena persistence dat při výpadku díky využití transakčního logu
- Dochází až k **30 násobnému zvýšení počtu zpracovaných transakcí** za vteřinu

# Vylepšení In-Memory OLTP

- Přidána podpora pro Computed Columns v memory optimized tabulkách
  - Včetně podpory indexů
- Přidání podpory následujících konstruktů v nativně kompilovaných modulech
  - CROSS APPLY
  - CASE
  - Práce s JSON v nativně kompilovaných modulech
- Odstranění limitu na počet indexů nad tabulkou (dříve 8)

# Resumable online index rebuild

- SQL Server 2017 umožňuje **pozastavit rebuild indexu** a následně v operaci pokračovat
- Možné využití
  - Rozdělení rebuildu rozsáhlého indexu do více navazujících údržeb databáze
  - Díky rozdělení operace není rebuild indexu v jedné rozsáhlé transakci, lepší správa volného místa v transakčním logu
  - Zotavení po selhání rebuildu indexu (došlo volné místo v DB)

```
ALTER INDEX PK_Person ON [Person].[Person] REBUILD  
WITH (ONLINE = ON, RESUMABLE = ON);
```

```
ALTER INDEX PK_Person ON [Person].[Person] PAUSE;
```

```
ALTER INDEX PK_Person ON [Person].[Person] RESUME;
```

# Online non-clustered columnstore index rebuild

- SQL Server 2017 přidává podporu pro **Online non-clustered columnstore index rebuild**

# Nové užitečné DMV

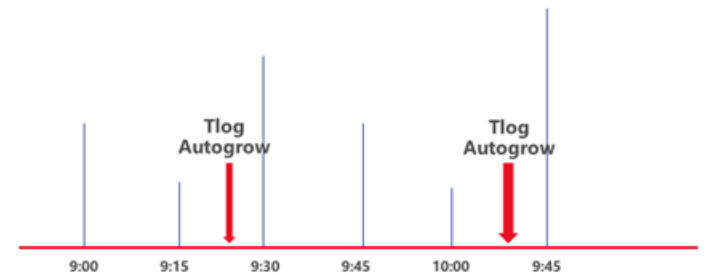
- **sys.dm\_db\_file\_space\_usage**
  - Nově modified\_extent\_page\_count
  - Využití k detekci a rozhodnutí, jestli je výhodnější Full Backup nebo Differential Backup
- **sys.dm\_tran\_version\_store\_space\_usage**
  - Statistiky využitého místa ve version store pro každou databázi

# Nové užitečné DMV

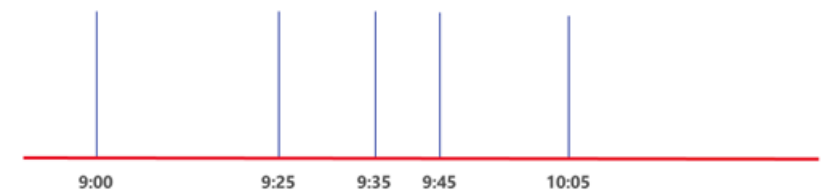
## ■ **sys.dm\_db\_log\_stats**

- Informace o stavu a struktuře transakčního logu
- DBCC LOGINFO
- Dle **log\_since\_last\_log\_backup\_mb** je možné spouštět zálohy logu tak, aby měly stejnou velikost

Backup every 15 mins irrespective of Tlog activity



Smart Transaction Log Backup



# SQL Server na Linuxu

- **SQL Server 2017 je možné provozovat na Linuxu**
  - Zákazníci s infrastrukturou kompletně na Linuxu mohou přejít na SQL Server bez nutnosti „komplikovat“ svou infrastrukturu Windows servery
- Jedná se o identický databázový engine jako na Windows, který je plně kompatibilní s Windows verzí
  - Interně se jedná o téměř totožné binární verze
- **Není zatím podporována kompletní funkcionality**
  - Problémy s funkcionalitou vázanou na Windows

# Omezení SQL Serveru na Linuxu

- Transactional replication, Merge replication
- Stretch DB, Polybase
- System extended stored procedures
- Filetable
- CLR assemblies s právy EXTERNAL\_ACCESS nebo UNSAFE
- Buffer Pool Extension



# Omezení SQL Server Agenta

- Nepodporované subsystemy
  - CmdExec, PowerShell, Queue Reader, SSIS, SSAS, SSRS
- Alerty
- Change Data Capture
- Log Reader Agent
- Managed Backup

# Chybějící komponenty SQL Serveru

- SQL Server Browser
- Analysis Services
- Reporting Services
- Data Quality Services
- Master Data Services
- SQL Server R services
- StreamInsight

# Co překvapivě funguje

- **Windows Authentication**

- SQL Server na Linuxu může být členem AD a používat Windows ověřování

- **Integration Services**

- Jsou podporovány v posledních vydáních

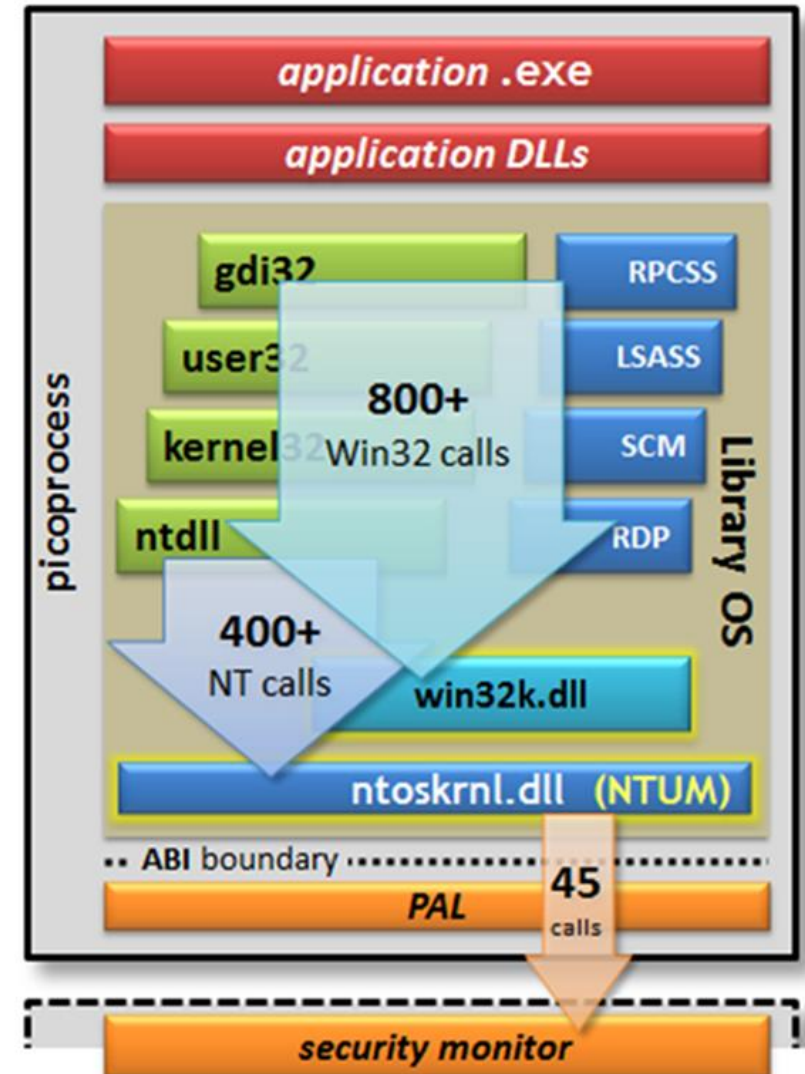
# Jak to funguje

- Od SQL Serveru 2005 je využívána abstrakce od OS Windows v podobě SQLOS
- SQLOS je však závislý na Windows
  - Win32 API, NT Kernel
- SQL Server je závislý na aplikačních knihovnách Windows
  - např. MSXML, CLR, COM
- SQL Server na Linuxu by měl být 100% kompatibilní, nebylo proto možné závislosti znovu implementovat pod Linuxem

# Jak to funguje

## ▪ Projekt **Drawbridge**

- Zaměřen na snížení spotřeby zdrojů při virtualizaci
- V rámci uživatelského procesu hostuje Windows Library OS
- LibOS konvertuje 1500+ Win API volání na 50 ABI volání pro interakci s hostitelem
- LibOS může hostovat další Windows komponenty požadované SQL Serverem
  - ♦ Nepotřebné části odstraněny (1% OS)
- Povedlo se tento picoprocess hostovat pod Linuxem



# Jak to funguje

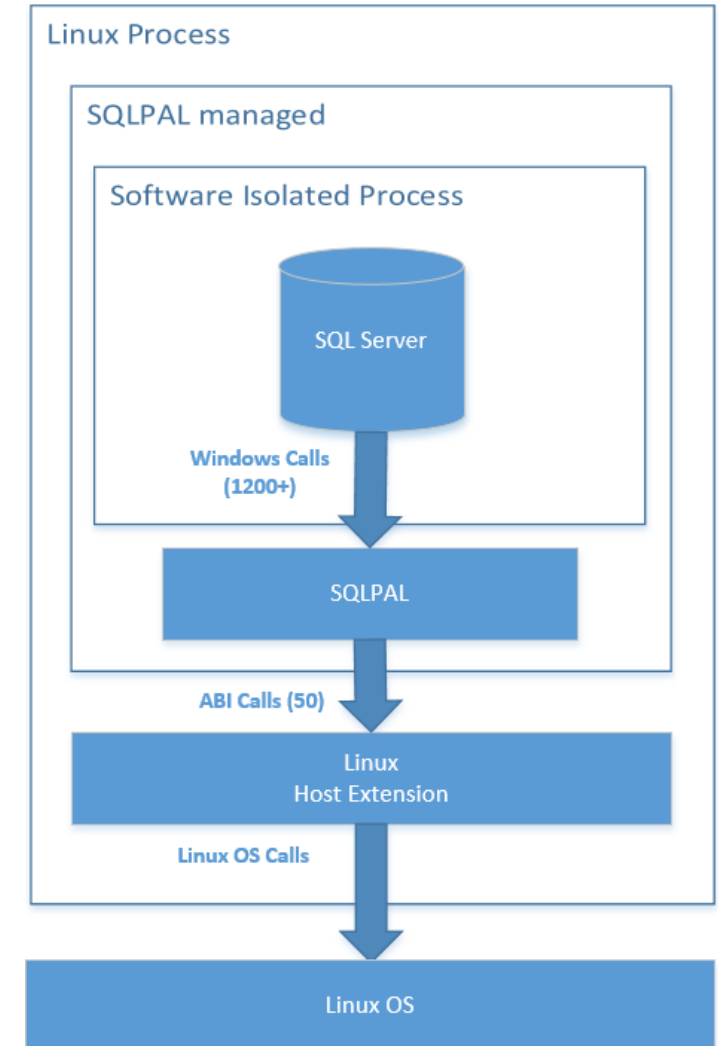
- Problém s efektivitou a duplicitními implementacemi

Technologies	SOS	Library OS	Host Extension
Object Management	✓	✓	✓
Memory Management	✓	✓	✓
Threading/Scheduling	✓	✓	✓
Synchronization	✓	✓	✓
I/O (Disk, Network)	✓	✓	✓

- Řešení: **SQL PAL (SQL Platform Abstraction Layer)**

# Jak to funguje

- SQL PAL (SQL Platform Abstraction Layer)
  - Nová vrstva mezi SQLOS a OS
  - Každé Win32 volání z SQL OS je abstrahováno
  - Pro kritická volání se obchází hostované Win32 API a NT Kernel a volání se provádí přímo proti ABI nad Linux Host Extension
  - Méně kritická volání jdou přes LibOS



# Instalace SQL Serveru pod Linuxem

- Instalace je závislá dle podporované distribuce Linuxu
- Red Hat Enterprise Linux 7.3
- SUSE Linux Enterprise Server v12 SP2
- Ubuntu 16.04
- Docker Engine 1.8+



# SQL Server v Dockeru

- Možnost zahostovat instalaci SQL Serveru v Dockeru
- Velmi zajímavé řešení pro DevOps
  - Možnost zkombinovat nasazení aplikace do kontejneru spolu s nasazením instance jejího databázového serveru
  - Vhodné spíše pro testovací scénáře, než produkční provoz SQL Serveru

# Clusterless AlwaysOn Availability Groups

- Availability Groups až doposud **vyžadovaly Windows Server Failover Cluster**
  - Řešení vysoké dostupnosti a zotavení po havárii
- Pokud předmětem nasazení **není** vysoká dostupnost, je možné nově použít Availability Groups za účelem škálování databáze – čtení ze sekundárních replik
- Zápis probíhá stále do jedné primární repliky

# Clusterless AlwaysOn Availability Groups

- Další podporovaný scénář je vytvoření Availability Groups mezi Windows a Linuxem
  - Na Linuxu nemáme Windows Server Failover Cluster
- Účelem může být migrace databází s minimálním výpadkem
  - Nastavíme Availability Group
  - Zreplikujeme databáze
  - Provedeme manuální fail-over
  - Zastavíme replikaci

# CREATE OR ALTER

- Od SQL Server 2016 SP1 je možné používat konstrukci CREATE OR ALTER pro tvorbu nebo změnu pohledu, uložených procedur nebo funkcí
  - Není třeba testovat existenci objektu a poté volit mezi CREATE/ALTER nebo nejdříve objekt odstranit

Chcete si o novinkách se mnou povídat  
4 dny a pořádně si je vyzkoušet?

Potom přijďte na moje školení [GOC 627 Novinky v Microsoft SQL Serveru 2016/2017 pro vývojáře a administrátory](#)

# Dotazy

**RNDr. David Gešvindr**

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

[david@wug.cz](mailto:david@wug.cz)

 @gesvindr