

riganti

 Update Conference
Prague 2020



Azure Cosmos DB

Optimalizace nákladů a change feed

Tomáš Herceg

CEO @ RIGANTI

Co-founder of Update Conference

Microsoft MVP

tomas.herceg@riganti.cz

[@hercegtomas](https://twitter.com/hercegtomas)

www.tomasherceg.com/blog



Úvod

- Globálně distribuovaná, vícemodelová databáze
- Podpora různých API
- Garantovaná nízká latence
- SLA
- Indexy spravovány na pozadí
 - Lze definovat i manuálně
- Multi-master writes



Podporovaná API

- SQL (Document DB)
 - Mongo DB
 - Graph (Gremlin)
 - Table (Azure Table Storage)
 - Cassandra
-
- API vybíráte při vytváření CosmosDB accountu

Azure Cosmos DB
New account

* ID
<enter a unique name>

* API **i**
SQL

* Subscription
<subscription ID>

* Resource Group **i**
 Create new Use existing
<enter same unique name as used in ID above>

* Location
<choose the location closest to users>

Enable geo-redundancy **i**

Pin to dashboard

Create Automation options



Základní poznatky

- Je to dokumentová databáze
 - Cosmos DB se vyvinula z Azure Document DB
- Efektivita dotazů má zásadní vliv na cenu
 - Rychlé (= levné) hledání podle primárního klíče
 - Rychlé (= levné) hledání v rámci partition
 - Jeden ze sloupců můžete nastavit jako partition key



Propustnost

- Jednotka RU/s
 - Každý dotaz má jinou náročnost
 - Lze zjistit v portálu
 - Čtení
 - 1kB dokument = 1 RU
 - 100kB dokument = 10 RU
 - Zápis
 - 1kB dokument = 5 RU
 - 100kB dokument = 50 RU
- Lze provisionovat na databázi (sdílené RU/s)
nebo na konkrétní kontejner



Ceny

- Standard provisioned throughput
 - Kolik si navolíte, tolik budete mít
 - Minimum 400 RU/s
 - Cca 23\$ / měsíc
 - Dříve bylo nutné mít 400 RU/s pro každou kolekci, ale lze je již sdílet
- Storage
 - Cca 0.25\$ / GB
- Platí se za každý region (pokud chcete škálovat)
- V případě multi-master writes, platí se víc (cca 2x tolik)



Ceny

- Autoscale provisioned throughput
 - Minimálně 4000 RU/s
 - Automaticky škáluje mezi 10 – 100% nastavených RU/s
 - Platíte jen za to, co využijete
 - Cena
 - 4000 RU/s – průměrné využití 10% – 35\$
 - 4000 RU/s – průměrné využití 100% – 350\$
- Free tier
 - Prvních 400 RU/s a 5GB místa je zdarma



DEMO

Interakce s Cosmos DB z .NETu



Kolik stojí dotazy

- Dotazování podle primárního klíče je levné
 - 2-3 RU (záleží na velikosti dokumentu, indexech atd.)
- Dotazování v rámci jedné partition
 - Relativně levné, obvykle jednotky RU
 - Záleží na velikosti partitions
- Dotazování napříč partitions
 - Většinou desítky RUs



Denormalizace

- V relačních databázích se typicky dodržují normální formy
 - Motivace
 - Ušetření místa na disku
 - Nejsou třeba hromadné updaty
 - Je jednodušší zajistit transakčnost
 - Nevýhody
 - Složité (a pomalé) dotazy s mnoha joiny
 - Čtení je typicky mnohem častější než zápis



Denormalizace

- Cosmos DB je stavěna na globální použití a multi-master writes
 - Tyto požadavky jdou proti sobě
- Datový model je jednodušší
 - JOINy se v podstatě nedělají
 - Dokumenty by měly obsahovat vše, co bude potřeba
 - Data jsou uložena v několika různých podobách
 - Klidně jen dočasně (TTL)
 - Eventuální konzistence



Praktický příklad

- Vlaky každou sekundu odesílají svou GPS polohu
 - Potřebujeme ukládat historii
 - → tabulka TrainPositions
- Požadavky
 - Na nádraží potřebujeme zobrazit tabuli příjezdů
 - Dispečer potřebuje vidět, kde vlaky aktuálně jsou
- Složité dotazy – seřazení od nejnovějších, dohledání posledního záznamu pro každý vlak...
 - V SQL Serveru by se použilo GROUP BY, PARTITION BY a ROW_NUMBER



Change Feed

TrainPositions

TrainId	Date	X	Y
Vlak A	12:34:56.322	12.23456	23.45678
Vlak B	12:34:56.467	54.45878	44.66872
Vlak A	12:34:57.497	12.23454	23.45682
Vlak B	12:34:57.601	54.45879	44.66875

...

Trains

TrainId	X	Y
Vlak A	12.23456	23.45678
Vlak B	54.45878	44.66872





Change Feed

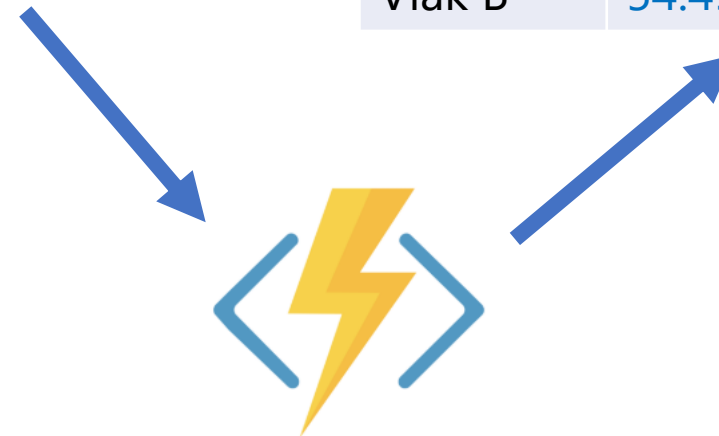
TrainPositions

TrainId	Date	X	Y
Vlak A	12:34:56.322	12.23456	23.45678
Vlak B	12:34:56.467	54.45878	44.66872
Vlak A	12:34:57.497	12.23454	23.45682
Vlak B	12:34:57.601	54.45879	44.66875

...

Trains

TrainId	X	Y
Vlak A	12.23454	23.45682
Vlak B	54.45879	44.66875





Change Feed

- Na insert či update záznamu je možné navázat další akci
 - Delete to neumí, nutno řešit pomocí soft-delete
- Jeden kontejner může mít více change feedů pro různé účely
- Je zachováno pořadí dle data změny v rámci jedné partition
 - Napříč partitions pořadí není – zpracovávají se paralelně
- Lze spustit od začátku pro všechny dokumenty v kontejneru



Eventuální konzistence

- Někdy časem se to provede
 - Polling s rostoucím intervalem, pokud se nic neděje
- Databáze dává změněný dokument do change feedu jen jednou
 - Pokud handler selže, je třeba retry logika – řeší přímo SDK
 - Handler se může zavolat více než jednou
- Více updatů jednoho dokumentu za sebou se může sloučit
 - V change feedu je vždy jen poslední verze dokumentu



Jak na to v .NETu

- Change Feed Processor Library
 - Řeší polling a spouštění handleru
 - Retry logika
 - Mnoho možností nastavení, vhodné pro složitější scénáře
- Azure Functions
 - CosmosDBTrigger umí funkci zavolat automaticky
 - Škáluje automaticky podle množství partitions
 - Jednoduché na použití
 - Interně používá Change Feed Processor Library



DEMO

Zpracování událostí z change feedu pomocí Azure Functions



Shrnutí

- Chovat se ke Cosmos DB jako k relační databázi se nevyplácí
 - Podpora Mongo DB a dalších API je tak trochu „marketing“
 - Funguje to, ale dotazy budou drahé a neefektivní
- Je to tak trochu „CQRS“
 - Zápis dokumentů v takové podobě, v jaké je máme
 - Change feed a funkce, které je transformují do podoby pro čtení
- Možnost nastavit na kontejneru TTL je užitečná



Update Conference Prague 2020

12 - 13 November 2020

Clarion Congress Hotel Prague

Save the Date!

updateconference.net

riganti

 Update Conference
Prague 2020



Q&A

Tomáš Herceg

CEO @ RIGANTI

Co-founder of Update Conference

Microsoft MVP

tomas.herceg@riganti.cz

[@hercegtomas](https://twitter.com/hercegtomas)

Microsoft takes .NET
Open Source

November 12th 2014