

# Key-value databázové systémy

Key-value database systems

Bc. Jan Jedlička

Bakalářská práce

Vedoucí práce: prof. Ing. Michal Krátký, Ph.D.

Ostrava, 2023

## **Abstrakt**

Cílem diplomové práce je popsat Key-value databázové systémy, ukázat výhodu těchto systémů a představit jedny z jejich významných představitelů. Součástí práce je návrh a implementace testovacího prostředí pro testování těchto systémů s ostatními SŘBD. Práce je zakončena vyhodnocením výsledků testů vybraných databázových systémů.

## **Klíčová slova**

NoSQL; Key-value databáze

## **Abstract**

The aim of the diploma thesis is to describe Key-value database systems, to show the advantage of these systems and to present some of their important representatives. Part of the work is the design and implementation of a test environment for testing these systems with other DBMS. The work is finished with an evaluation of the test results of selected database systems.

## **Keywords**

NoSQL; Key-value database

## **Poděkování**

Rád bych na tomto místě poděkoval vedoucímu diplomové práce panu prof. Ing. Michalovi Krátkému, za pravidelné konzultace a poskytnutí mnoha užitečných rad a nápadů pro řešení samotné práce.

# Obsah

<b>Seznam použitých symbolů a zkratek</b>	<b>5</b>
<b>1 Úvod</b>	<b>6</b>
<b>2 NoSQL Key-value databázové systémy</b>	<b>7</b>
2.1 Amazon DynamoDB . . . . .	7
2.2 Oracle NoSQL Database . . . . .	7
2.3 Redis . . . . .	8
2.4 Aerospike . . . . .	8
2.5 Oracle Berkeley DB . . . . .	9
2.6 Riak KV . . . . .	9
2.7 Voldemort . . . . .	10
2.8 InfinityDB . . . . .	11
2.9 Porovnání KV DB . . . . .	12
2.10 Nezmíněné významné KV DB v 2022 . . . . .	13
<b>3 Prostředí pro testování databázových systémů</b>	<b>14</b>
<b>4 Vyhodnocení výsledků testů</b>	<b>15</b>
<b>5 Závěr</b>	<b>16</b>
<b>Literatura</b>	<b>17</b>

# Seznam použitých zkratek a symbolů

NoSQL	– No Structured Query Language
Key-value database	– Klíč-hodnota databáze
TTL	– Time to live

# Kapitola 1

## Úvod

NoSQL Key-value (neboli Klíč-hodnota) databáze[1] je jedno z paradigmat pro uložení dat. Databáze je navržena pro ukládání, načítání a správu různých datových struktur, dnes známých jako slovníky nebo hashovací tabulky. Slovníky obsahují kolekci objektů či záznamů, které mohou opět obsahovat množinu různých polí s daty. Záznamy jsou do slovníků, či hashovacích tabulek, ukládány za pomoci klíče, který identifikuje pozici záznamu v datové struktuře a používá se k následnému vyhledávání dat v databázi.

Key-value databáze fungují velice odlišně než tradiční relační databázové systémy. Relační databáze mají předdefinovanou datovou strukturu v databázi jako sérii tabulek s dopředu definovanými datovými typy. Díky tomuto modelu může relační databázový systém provádět řadu optimalizací. Na druhou stranu Key-value databázové systémy mohou mít pro každý záznam různě definované kolekce dat s odlišnými velikostmi a počty atributů. Tato vlastnost nabízí Key-value databázovým systémům flexibilitu a možnost přiblížení se k objektově orientovanému programování. Protože Key-value databáze nevyžaduje pevně nastavené datkové typy hodnot, jako je tomu u relační databáze, tak Key-value databáze často potřebují méně paměti k uložení stajných dat, což může vést k značnému nárůstu výkonu. Dále tyto databáze bývají distribuované a dosahují horizontální až lineární škálovatelnosti.

Výkon a nedostatečná standardizace omezovaly Key-value databázové systémy pouze na specializovaná využití, ale díky rychlému přechodu na cloud computing dochází v posledních letech k rozšíření obecné využitelnosti NoSQL databázových systémů. Například databázový systém Redis[2] je v současnosti jedním z deseti nejlépe hodnocených[3] databázových systémů napříč relačními i NoSQL databázovými systémy.

## Kapitola 2

# NoSQL Key-value databázové systémy

V současné době existuje nespočet různých Key-value databázových systémů, od malých open source projektů po velké placené cloud služby. Různé systémy disponují odlišnými vlastnostmi jako je propustnost, škálovatelnost, uživatelská přívětivost skrz dotazovací jazyk a podporu aj. Dle průzkumu[4][5][3] bylo vybráno 10 aktuálně významných Key-value databázových systémů se snahou o jednoduchý popis, porovnání a konečný výběr vhodných Key-value databázových systémů se snahou otestovat vlastnosti těchto systémů.

### 2.1 Amazon DynamoDB

Amazon DynamoDB[6] je v současné době největší a nejvyžívanější Key-value databázový systém. Jedná se o serverless cloud systém s odezvou v řádu jednotek mikrosekund a využitím v oblastech jako je web tech, IoT, mobile a gaming. DynamoDB je plně a automaticky spravovatelná, multi master databáze zaměřená na vysoké využití horizontální škálovatelnosti. Unikátní primární klíče umožňují identifikaci jednotlivých záznamů v tabulkách a sekundární index zlepšuje dotazovací flexibilitu. Primární klíč je jako vstup do hashovací funkce a výsledky hash nám udává fyzickou pozici uloženého záznamu. DynamoDB poskytuje silnou konzistenci na čtení hodnot od posledních aktualizací. Atomické čítače umožňují automatické změny hodnot číselných atributů. Využívá TTL pro prošlé záznamy v tabulkách. Archivace dat je umožněna díky full backupu. Systém má konzolové API pro správu databáze a práci s daty, systém však nabízí i možnost využití SQL jazyka. Amazon DynamoDB má i VPC pro soukromou komunikaci bez potřeby využití internetu.

### 2.2 Oracle NoSQL Database

Oracle NoSQL Database[7] je databázová cloud služba vhodná pro práci s velkými objemy dat a odhadovatelnou nízkou odezvou v řádu jednotek milisekund. Služba je postavena na enginu z Oracle Berkeley DB. Databáze je plně spravovatelná, flexibilní, škáluje horizontálně, dynamicky a dosahuje

vysokých výkonů. Mimo Key-value data se jedná i o spolehlivé uložení pro dokumenty a data s pevně daným schématem. Vzhledem k tomu že databázový systém je plně spravovaný společností Oracle, tak je pro vývojáře rychlé a snadné začít tuto službu využívat a soustředit se pouze na vývoj aplikací, neboť není potřeba se obtěžovat se správou základní infrastruktury databáze, softwaru, zabezpečení atp. Jedná se o Single Master, Multi Replica grafový systém. Pokud dojde k chybě na masteru, je master automaticky nahrazen jednou z replik. Pro Key-value ukládání s kapacitu jednotek terabytů využívá systém velký počet Storage uzlů, které je možno skupinově konfigurovat. Pro udržení konzistence jsou Storage uzly replikovány. Uzly a hrany v grafu reprezentují entity které vytvářejí vztahy a propojení. Sdílený systém, uniformně alokuje data okolo ostatních částí skupin. Databáze obsahuje i SQL Query s jazykem pro import, export a přenos dat mezi různými Oracle NoSQL databázemi. Mimo jiné je zde podpora i pro Failover, SwitchOver, Bulk Get API, Off Heap Cache a podpora Big Data SQL.

## 2.3 Redis

Redis[2] je in-memory uložení pro datové struktury, využívané jako Key-value databáze, cache, streaming engine nebo zprostředkovatel zpráv. Toto datové uložení má skvělé využití pro klíče v podobě hashe a hodnoty jako velký json objekt. Pro perzistenci dat můžeme ukládání dat na disk provádět po nasatvitelných pravidelných intervalech, nebo je možné data logovat vždy při vykonávání operací, pokud nemáme zájem o trvanlivost dat je možné ukládání dat vypnout úplně a datové uložení využít čistě jako cache. Uložení škáluje horizontálně. Redis podporuje datové struktury jako stringy, hashe, listy, množiny, bitmapy, hyperloglog a geospatial indexy. Nad datovými typy Redis umožňuje rychlé atomické operace jako rozšíření stringů, přidání prvků na začátek a konec listů atd. Datové uložení také poskytuje setříděné množiny pro vytváření indexů dle ID nebo jiného číselného atributu. Redis hashing ukládá data jako klíč a mapu. Keyspace notifikace dovoluje klientům odebírat Publisher-Subscriber kanály. Pro práci s dotazy na souřadnice a geometrii je možné využívat Geo API. Redis umožňuje provádět transakce, volat lua skripty a nastavovat různé úrovně TTL pro záznamy. Redis podporuje Trivial-to-setup Master-Slave asynchronního replikování, společně s rychlou neblokující se prvotní synchronizací. Struktura pro ukládání dat je Single-rooted replikovaný strom. Redis má vlastní API pro práci s daty pro populární programovací jazyky jako C, Python, Java a Javascript.

## 2.4 Aerospike

Aerospike[8] je Key-value databáze využívající Hybrid Memory architekturu[9], která umožňuje odezvu do jednotek milisekund a vysokou propustnost v řádu stovek tisíc až milionu operací za sekundu. Hybrid Memory architektura od Aerospike je implementována tak, že index je čistě In-Memory, tím pádem není index persistentní (vhodné například pro uživatelské cache sessions), a data jsou uložena



čistě persistentně na SSD disku a čtou se přímo z něj. Díky tomu že je Aerospike jako Key-value databáze naprosto Schema-less tak je možné definovat Sets a Bins za běhu pro maximální flexibilitu aplikací. Databáze škáluje lineárně a poskytuje silnou konzistenci, nízkou cenu a korektnost. Umožňuje real-time analýzu pro rychlé rozhodování a dynamickou optimalizaci pro vhodné využívání zdrojů dat, proto je databáze vhodná pro velké a stále aktualizované databáze. Poskytuje server-side clustering, bezpečnost na transportní vrstvě. Databáze také umožňuje Customer deployment s nulovým downtime. V praxi se Aerospike díky svým vlastnostem využívá například pro banking, telekomunikace, adtech a gaming. Aerospike poskytuje vlastní silný dotazovací jazyk AQL[10], který má prakticky shodnou syntaxi jako SQL (i když se o SQL nejedná). vlastní vytvořitelné agregační funkce pomocí Lua jazyku (flexibilní pro agregační algoritmy).

## 2.5 Oracle Berkeley DB

Oracle Berkeley DB[14] je rodina vestavěných Key-value databázových knihoven. Jedná se o čistě In-memory databázi, díky čemuž dosahuje vysokého výkonu a odezvy v jednotkách mikrosekund. Databáze škáluje horizontálně. Data jsou replikována pro vysokou dostupnost z vícero zdrojů a dobrou toleranci chybovosti. Oracle Berkeley DB využívá vhodné datové struktury pro práci s daty jako je B-strom, hash table indexy nebo fronta. Databáze využívá obnovitelné ACID transakce a poskytuje několik různých úrovní izolace (včetně MVCC[11]). Data jsou dělena do oddílů dle key ranges. Umožňuje komprimaci dat. Databáze je Single-master, Multi-replica, tedy je vysoce dostupná a umožňuje dobrou konfigurovatelnost. Repliky umožňují čtecí škálovatelnost, rychlý fail-over, hot-standby a další distribuované konfigurace dodávající podnikové prostředky v malém, vestavěném balíčku. Pro přístup k datům a nastavení databáze se využívá jednoduché volání function-call API. Spousta moderních programovacích jazyků, jako například C++, C#, Javy, Python atp. podporuje tyto knihovny. Data mohou být ukládána v nativním formátu aplikace, XML, SQL nebo jako Java Objekty. Oracle Berkeley DB je vhodný nástroj pro vše od lokálního uložení po world-wide distribuovanou databázi (od kilobytů po petabyty).

## 2.6 Riak KV

Riak KV[12] je distribuovaná NoSQL Key-value databáze s pokročilou lokální a multi-cluster replikací, která garantuje čtení a zápis i v případě selhání hardwaru nebo síťových oddílů.

- garance čtení a zápisu i při selhání HW nebo síťových oddílů
- bez konfliktů replikované datové typy (CRDTs), flags, registry, čítače, množiny a mapy
- konfigurace aktivního clusteru, poskytování dat pro klienty díky nízké latenci z nejbližšího data centra

- dostupné zóny, multi cluster repliky a redundance dat v geografickém regionu
- flexibilní datový model bez pre definovaného schématu
- vylepšené logování chyb a reporty
- automatická komprese dat pomocí Snappy kompresní knihovny
- vhodná pro ukládání velkého množství nestrukturovaných dat
- pro big-data aplikace, ukládání dat z připojených zařízeních a replikování dat do okolí
- automatická distribuce dat skrz cluster, pro robustnost, výkon
- master-less architektura, vysoká dostupnost, téměř lineární škálovatelnost za využití snadného přidání HW kapacity bez nutnosti mnoha operací
- nízká latence, vhodná pro chat/messaging aplikace
- možnost zpracování dat pro získání užitečných závěrů a akceschopných informací
- design pro horizontální škálovatelnost s komoditním HW, jednoduché pro rozšíření objemu dat bez potřeby vytváření komplexního sdílení
- bez vnucování restrikcí na hodnoty, session data mohou být enkódována mnoha způsoby a nevyžadují změnu schématu
- během nejvyšší zátěže nezhoršuje zápis a horizontální škálovatelnost, uživatelé jsou obslouženi bez problémů
- dobrý pro soukromé, veřejné i hybridní cloud nasazování

## 2.7 Voldemort

- distribuovaná kv db založena na Amazon DynamoDB
- automatická replikace dat skrz více serverů
- automatické rozdělování dat mezi servery, každý server obsahuje pouze část z celkových dat
- nastavitelná konzistenčnost
- transparentní ošetřování chyb serverů
- zapojitelný storage-engine (MySQL, Read-Only) a serializace (Java Serialization, Thrift, Avro)
- verzování dat pro maximální integritu i během poruch

- každý uzel je samostatný a nezávislý, žádný centrální řídicí uzel nebo uzel řídicí řešení chyb
- dobrá výkonost na jeden uzel, 10-20 tisíc operací za sekundu (1 op. za 50 mikro sekund) dle HW, sítě, systému disku atp.
- podpora zapojitelné strategie pro rozpoložení dat, pro možnost distribuce dat skrz data centra která jsou mezi sebou geologicky velice vzdálená
- využívá JMX pro zlepšení viditelnosti pro interní monitorování a validaci dat
- In-Memory caching pro eliminaci oddělených částí cache, jednoduché a rychlé in-memory tenstování (unit testy)
- horizontální škálování čtení i zápisu
- API rozhoduje o replikování a místě ukládání dat, různé strategie pro specifické aplikace
- široké možnosti pro klíče i hodnoty díky serializaci, listy a tuply s pojmenovanými poli
- JSON data model pro serializaci ale v kompaktním bytovém formátu, typová kontrola dat dle očekávaného schématu
- hashovatelné schéma, vyhledávání dle primárního klíče a možnost modifikace jednotlivých hodnot
- jednoduchá distribuce skrz stroje protože data mohou být rozdělována dle primárních klíčů
- dostupnost a bezpečnost jednotlivých oddílů při vysoké propustnosti

## 2.8 InfinityDB

- Nestable Multi-value, může reprezentovat stromy, grafy, Key/Value mapy, dokumenty, velká řidká pole, tabulky
- ACID pro vlákna, ACD pro bulk operace
- jednoduché API, instantní produktivita developerů
- dynamický pohled dotazů (set logic views, delta views, ranges)
- samostatná administrace, jeden soubor (bez konfiguračních souborů, logů, dočasných souborů, upgrade skriptů a DBA)
- není potřeba dělat čištění junk souborů po operacích když zde nejsou žádné zanechány
- runtime schema evolution (pro dopřednou a zpětnou kompatibilitu)

- all-Java db pro servery, pracovní stanice, ruční zařízení
- protokol robustní vnitřní uložení pro vytrvání na požádání nebo rozdělení cache na disk pro velké množství dat
- single data file, aktuální, bezpečný, korektní a využitelný pro každý případ (je designovaný pro použití jednoho souboru)
- bez zotavení na základě logu, restart a zotavení je vždy okamžité
- kombinace jednoho souboru a instantního zotavení nevyžaduje administraci db
- prostor pro uložení strukturovaných, polostrukturovaných a nestrukturovaných dat, tento jednoduchý model umožňuje ukládání stromů, grafů apod.
- možnost využít In-Memory-Only db která nechá všechna data v cache, nebo naopak se data ukládají normálně do souboru, je možnost si vybrat bez změny kódu
- přístup k datům v cache je plně více vláknový
- data která nejsou často využívána jsou stránkována na disk

## 2.9 Porovnání KV DB

Název	Správa	Škálovatelnost	Odezva	Zotavení
Amazon DynamoDB	automatická, plně spravovatelná	vysoká, horizontální	mikrosekundy	logy, záloha, automatické obnovení
Oracle NoSQL DB	nízká	horizontální	milisekundy	replika je prohlášena za master
InfinityDB	jeden soubor se vším	nízká	milisekundy	bez logů, okamžité ale ztrácíme data
Redis	plná	horizontální	milisekundy	z logů (logování snižuje výkon)
Aerospike	plně spravovatelná	lineární	milisekundy	logy, záloha
Oracle Berkeley DB	velká	horizontální	mikrosekundy	repliy
Riak KV	vysoká	horizontální, téměř lineární	milisekundy	multi cluster repliky, logování
Voldemort	vysoká	horizontální	milisekundy	repliky

## **2.10 Nezmíněné významné KV DB v 2022**

- MongoDB
- Couchbase
- Azure Cosmos DB

## Kapitola 3

# Prostředí pro testování databázových systémů

TODO

## Kapitola 4

# Vyhodnocení výsledků testů

TODO

## Kapitola 5

# Závěr

TODO



# Literatura

1. *Key-value database* [online]. 2022. [cit. 2022-11-18]. Dostupné z: [https://en.wikipedia.org/wiki/Key%E2%80%93value\\_database](https://en.wikipedia.org/wiki/Key%E2%80%93value_database).
2. *Redis* [online]. 2022. [cit. 2022-11-18]. Dostupné z: <https://redis.io/>.
3. *DB-Engines Ranking* [online]. 2022. [cit. 2022-11-18]. Dostupné z: <https://db-engines.com/en/ranking>.
4. *Top NoSQL Key Value store Databases: Predictiveanalyticstoday* [online]. 2022. [cit. 2022-11-13]. Dostupné z: <https://www.predictiveanalyticstoday.com/top-sql-key-value-store-databases/>.
5. *Best Document Databases: G2* [online]. 2022. [cit. 2022-11-13]. Dostupné z: <https://www.g2.com/categories/document-databases>.
6. *Amazon DynamoDB* [online]. 2022. [cit. 2022-11-18]. Dostupné z: <https://aws.amazon.com/dynamodb/>.
7. *Oracle NoSQL Database* [online]. 2022. [cit. 2022-11-19]. Dostupné z: <https://www.oracle.com/database/nosql/technologies/nosql/>.
8. *Aerospike* [online]. 2022. [cit. 2022-11-20]. Dostupné z: <https://aerospike.com/>.
9. *Hybrid Memory Architecture* [online]. 2022. [cit. 2022-11-20]. Dostupné z: <https://aerospike.com/products/features/hybrid-memory-architecture/>.
10. *Aerospike Quick Look (AQL)* [online]. 2022. [cit. 2022-11-20]. Dostupné z: <https://docs.aerospike.com/tools/aql>.
11. *Multiversion concurrency control* [online]. 2022. [cit. 2022-11-21]. Dostupné z: [https://en.wikipedia.org/wiki/Multiversion\\_concurrency\\_control](https://en.wikipedia.org/wiki/Multiversion_concurrency_control).
12. *Riak KV* [online]. 2022. [cit. 2022-11-21]. Dostupné z: <https://riak.com/products/riak-kv/index.html>.
13. *How do NoSQL databases work? Simply Explained! youtube* [online]. 2021. [cit. 2022-11-18]. Dostupné z: <https://www.youtube.com/watch?v=0buKQHokLK8>.

14. *Oracle Berkeley DB* [online]. 2022. [cit. 2022-11-21]. Dostupné z: <https://www.oracle.com/database/technologies/related/berkeleydb.html>.