

1 NoSQL Key Value databázové systémy

1.1 Amazon DynamoDB

- serverless databáze
- odezva microsekundy
- web tech, IoT, mobile, gaming
- plně spravovatelná multi master databáze
- odhadovatelný výkon a bezproblémová škálovatelnost
- unikátní primární klíč pro identifikaci jednotlivých záznamů v tabulce
- sekundární index pro lepší dotazovací flexibilitu
- primární klíč je jako vstup do hash funkce, výsledný hash je fyzická pozice uloženého záznamu
- silná konzistence na čtení hodnot od poslední aktualizace
- atomic counters pro automatické změny hodnot číselných atributů
- TTL pro prošlé záznamy v tabulkách
- full backup pro archivaci dat
- automatická správa systému
- DynamoDB konzolové api pro správu db
- VPC pro soukromou komunikaci bez potřeby využití internetu

1.2 Oracle NoSql Database

- vhodná pro velké objemy dat a nízkou odezvu
- velký počet storage uzlů pro lepší výstup a kapacitu uložení
- díky DB Java Edition high-availability storage engine
- single master, multi replica - vysoká dostupnost a nízká chybovost
- při chybě na masteru je jedna z replik automaticky prohlášena za nový master
- grafová databáze
- integrace v různých Oracle a Open Source aplikacích
- spolehlivá a flexibilní správa konfigurace skupiny uzlů

- škálovatelná se spolehlivostí pro poskytnutí správy všech dat pro zajištění spolehlivosti
- uzly a hrany v grafu reprezentují entity které vytvářejí vztahy a propojení
- sdílený systém, uniformně alokuje data okolo ostatních částí skupin
- storage uzly jsou replikované pro udržení konzistence
- obsahuje SQL Query s jazykem pro import, export a přenos dat mezi různými Oracle NoSQL databázemi
- Failover, SwitchOver, Bulk Get API, Off Heap Cache a podpora Big Data SQL a Apache Hive integration

1.3 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, roční zařízení
- protokol robustní vnitřní uložště 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 cachy, nebo naopak se data ukládají normálně do souboru, je možnost si vybrat bez změny kódu
- přístup k datům v cachy je plně více vláknový
- data která nejsou často využívána jsou stránkována na disk

1.4 Redis

- in-memory uložště pro datové struktury, využívána jako db, cache nebo zprostředkovatel zpráv
- můžeme jednou za čas ukládat pravidelně data na disk, nebo provádět logování při provádění operací (nemusíme nic ukládat a mít čistě in memory db)
- podporuje datové struktury jako stringy, hashe, listy, množiny, bitmapy, hyperloglog a geospatial indexy
- Keyspace notifikace dovoluje klientům odebírat Pub/Sub kanály
- setřízené množiny pro vytváření indexů dle ID nebo jiného číselného atributu
- Geo APi pro dotazy na souřadnice nebo rádiusy
- Radius hashing ukládá data jako klíč a mapu
- Single-rooted replikovaný strom
- API pro populární používané jazyky
- obsahuje transakce, lua skripty a různé úrovně trvání na disku
- atomické operace (rozšíření stringu, přidání prvku do listu atp.)
- podpora trivial-to-setup master-slave asynchronního replikování
- rychlá neblokující se prvotní synchronizace, automatické znovu připojování se znovu synchronizací na netsplit
- skvělé využití pro Key = hash a Value = velký json objekt

1.5 Aerospike

- architektura hybrid memory
- internet scale, odhadovatelná a vysoká výkonost
- korektnost, silná konzistence, nízká cena, lineární škálovatelnost
- real-time rozhodování na velké, stále aktualizované db

- dynamická optimalizace pro optimální využití zdrojů
- server-side clustering
- bezpečnost na transportní vrstvě
- banking, telekomunikace, adtech, gaming
- customer deployment s zero downtime
- transakce v milisekundách
- výběr in-memory storage (cache sessions) nebo SSD disk storage (pro trvanlivost dat) bez kompromisů ve výkonnosti
- silný query jazyk, vlastní vytvořitelné agregační funkce pomocí Lua jazyku (flexibilní pro agregační algoritmy)
- schema-less, sets/bins mohou být přidány za běhu (maximální flexibilita pro aplikace)

1.6 Oracle Berkeley DB

- knihovna pro službu key-value db
- čistě in-memory db
- ultra-low odezva, škálovatelná, vysoce výkonná
- velice dostupná, tolerance chybovosti
- B-stromy, fronty, hash data indexy
- obnovitelné ACID transakce, více úrovní izolace (včetně MVCC)
- rozdělení dat dle key ranges
- možnost komprese záznamů
- jednoduché volání API pro přístup k datům a nastavení db
- stavební části pro db od lokálního uložení po world-wide distribuovanou db (od KiloBytů po PetaByty)
- data uložená v XML, SQL (když není potřeba tak se využívá právě Key-Value uložení), Java Objekty
- podpora moderních programovacích jazyků (C++, C#, Java, Python atp.)
- Single master, multi replica, vysoce dostupná konfigurovatelnost db
- repliky umožňují čtení škálovatelnost, rychlý fail-over, hot-standby a další distribuované konfigurace dodávající podnikové prostředky v malém, vestavěném balíčku

1.7 Riak KV

- distribuovaná kv db, pokročilé lokální a multi-cluster replikování
- 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 předdefinovaného 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 zařízení 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

1.8 Project 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