



NoSQL DATABÁZE

Lukáš Matějů

27.2.2024 | DPB



ČÁST I.: OPAKOVÁNÍ



OPAKOVÁNÍ

- **big data**

„Soubory dat, jejichž velikost je mimo schopnosti zachycovat, spravovat a zpracovávat data běžně používanými softwarovými prostředky v rozumném čase.“ [Gartner]

- **éra big data**
 - stálý a rychlý přísun nových dat
 - cloud computing
 - dynamická a škálovatelná analýza dat
- **aplikace**
 - cílený marketing, doporučovací systémy
 - analýza sentimentu, mobilní reklamy
 - chování skupiny, ...



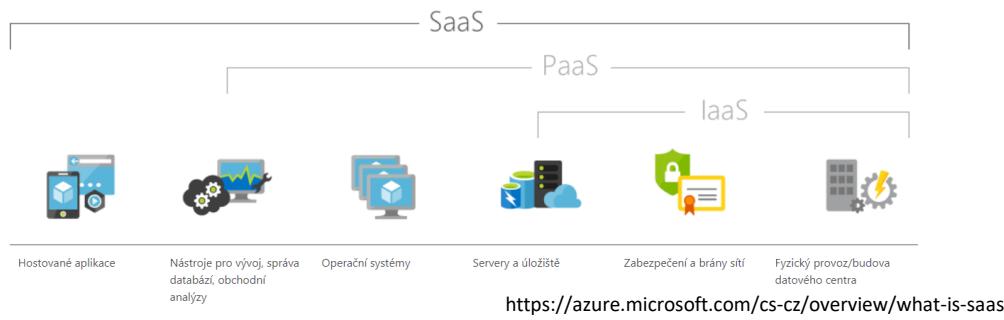
OPAKOVÁNÍ

- big data
 - dělení
 - strojová, lidská, organizační
 - strukturovaná, částečně strukturovaná, nestrukturovaná
 - hlavní přínos v kombinaci
 - charakteristika
 - 3 základní V's
 - objem (volume), různorodost (variety), rychlosť (velocity)
 - 2+1 doplňkové V's
 - věrohodnost (veracity), valence (valence), hodnota (value)
 - kam s nimi?
 - NoSQL databáze



OPAKOVÁNÍ

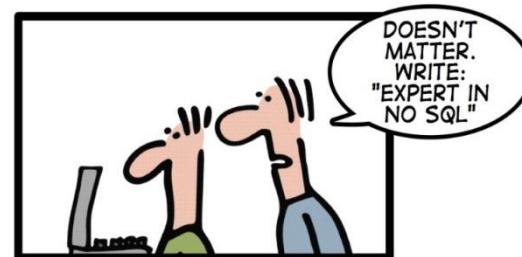
- **cloud computing**
 - internetový model vývoje a používání počítačových technologií
 - dostupnost systémových prostředků na vyžádání bez aktivní správy klienta
 - nejčastěji datová (cloudová) úložiště a výpočetní výkon
 - systémové prostředky hostované na vzdálených datových centrech
 - spravované cloudovými poskytovateli služeb
 - uživatelé k nim přistupují přes internet
 - prostředky jsou uživatelům pronajímány
 - distribuční model
 - infrastruktura, platforma, software jako služba



ČÁST II.: NoSQL DATABÁZE



HOW TO WRITE A CV



Leverage the NoSQL boom

<http://geek-and-poke.com/>

NoSQL DATABÁZE

- význam NoSQL
 - často chybně – no SQL (žádné SQL)
 - správně – not only SQL (ne jen SQL)
 - u některých databází je možné SQL používat
- nerelační databáze
 - žádné relace (tabulky)
- technologie známá již od 60. let minulého století
 - snaha vyřešit problémy relačních databází (jaké jsou?)
 - jedno řešení pro všechno
- ale až v současné době prudký nárůst popularity
 - obrovské objemy a široká škála dat
 - generované cloudy, mobilními zařízeními, sociálními médií, ...



NoSQL DATABÁZE

➤ NoSQL databáze se vyvinuly

- dokáží zpracovávat obrovské objemy rychle se měnících nestrukturovaných dat
- pomáhají vývojářům rychle vytvářet databázové systémy pro ukládání nových informací a jejich přípravu pro vyhledávání a analýzu
- umožňují
 - flexibilní vývoj
 - flexibilní zpracování dat
 - provoz v libovolném měřítku

➤ cílem tedy ve výsledku není nahradit relační databáze

- jedná se spíš o alternativu s jinou aplikací (kladivo vs. sekera)
- vhodné např. pro big data a webové aplikace v reálném čase



NEJVHODNĚJŠÍ PRO

- relační databáze
 - zpracování relačních dat majících logické a diskrétní požadavky určitelné předem
 - schéma potřebné udržovat a synchronizovat mezi aplikací a databází
 - starší systémy vytvořené pro relační struktury
 - aplikace vyžadující komplexní dotazování nebo transakce s více řádky
- NoSQL databáze
 - zpracování velkých objemů vzájemně nesouvisejících, nebo rychle se měnících dat
 - data nezávislá na schématu nebo schéma určené aplikací
 - aplikace preferující výkon a dostupnost než silnou konzistenci
 - neustále přístupné aplikace sloužící uživatelům po celém světě



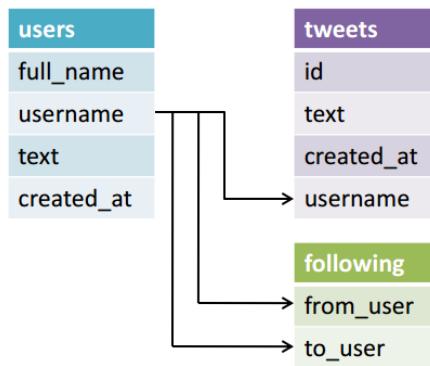
TYP DATABÁZE

- relační databáze
 - definované relace (tabulky)
 - definované atributy (sloupce)
 - definované vazby mezi relacemi
 - data ukládána ve formě záznamů (řádků) v relacích
 - související data jsou uložena odděleně
 - v případě dotazů spojovány
 - umožňují snadno modelovat vztahy mezi daty
- NoSQL databáze
 - nerelační databáze
 - žádné relace
 - definovaný mechanismus pro ukládání a načítání dat
 - liší se databázi od databáze
 - vyhýbají se vazbám mezi daty
 - distribuované databáze



TYPY DATABÁZÍ

- relační databáze
 - databáze relací záznamů seskupených do vztahů
- NoSQL databáze
 - key-value úložiště
 - dokumentové databáze
 - sloupcová úložiště
 - grafová úložiště
 - databáze časových řad
 - a další ...



<http://www.padakuu.com/article/295-relational-databases>



DATOVÉ SCHÉMA

- relační databáze
 - strukturovaná data
 - pevně předdefinované datové schéma
 - jasně definuje databázi
 - určuje relace, atributy, vztahy
 - ale také např. datové typy, indexy, pohledy, procedury, triggery a veškerá další omezení
- NoSQL databáze
 - často umožňují všechna data
 - strukturovaná data
 - částečně strukturovaná data
 - nestrukturovaná data
 - různé typy big data
 - volné schéma
 - žádné předdefinové nastavení
 - flexibilita



HIERARCHICKÉ UKLÁDÁNÍ DAT

- relační databáze
 - obtížné hierarchické ukládání
 - data ukládána v relacích
 - s nárůstem počtu relací roste i náročnost na vytváření vazeb
 - udržitelné jen do určitého bodu
- NoSQL databáze
 - snadné hierarchické ukládání
 - klíč – hodnota
 - vhodné pro velké objemy dat
 - big data

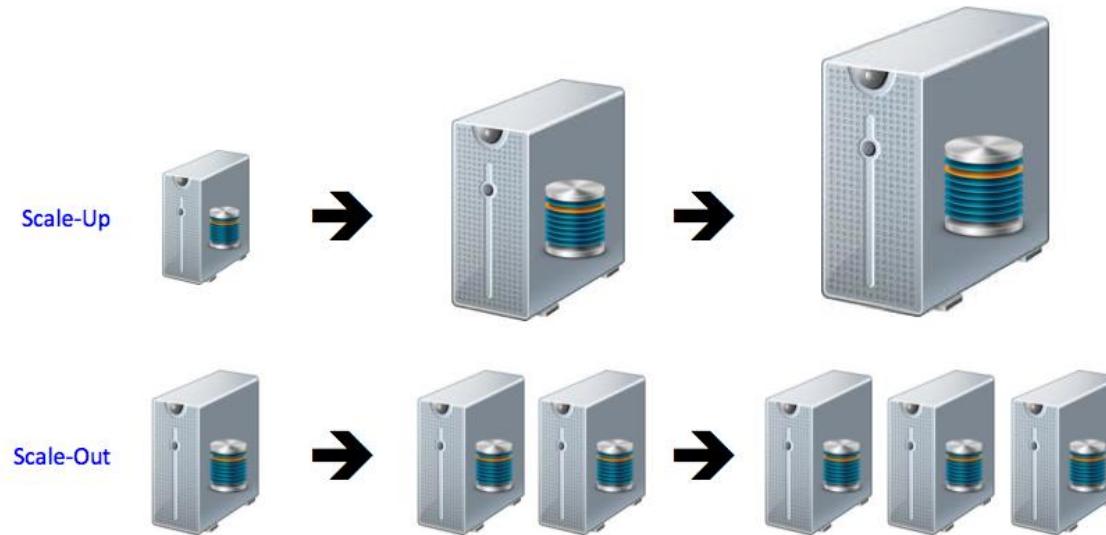


ŠKÁLOVÁNÍ

- škálovatelnost je vlastnost systému zvládat rostoucí množství požadavků přidáváním dalších prostředků
- relační databáze
 - vertikální škálování (nahoru)
 - změna vlastností stávajících prvků
 - přidání výpočetní síly
 - zpracování dat se nemění
 - je prováděno na výkonnějším stroji
 - zvýšení kapacity
 - nové komponenty jsou drahé
 - disky, paměti, CPU, sítování, atd.
 - nelze donekonečna
 - jednoduchost
- NoSQL databáze
 - horizontální škálování (ven)
 - přidání (nebo ubrání) prvků
 - přidání výpočetních uzelů
 - přesun k paralelnímu zpracování
 - rozdělení dat mezi uzly – sharding
 - zvýšení kapacity
 - nové komponenty jsou levné
 - základní HW
 - clustery
 - výkon, flexibilita, redundance



ŠKÁLOVÁNÍ



<https://medium.com/faun/scalability-248019b918ed>

ONLINE ZPRACOVÁNÍ

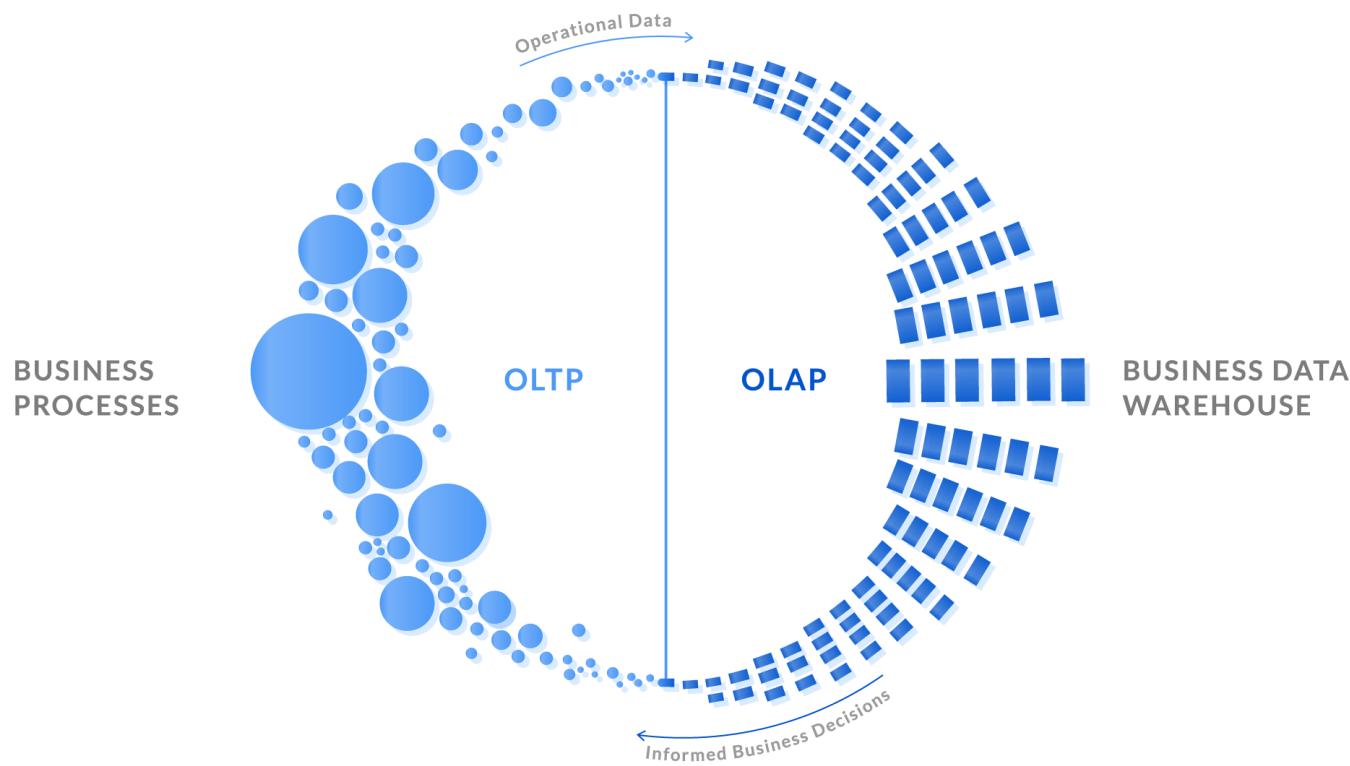
- relační databáze
 - vhodné pro aplikace s velkým množstvím transakcí
 - transakce se provede celá, nebo vůbec (ACID)
 - OLTP
 - online transaction processing
 - softwarové nástroje s podporou transakcí
 - ukládání a správa dat pro každodenní operace
 - správa každodenních transakcí
 - velké množství online transakcí
 - NoSQL databáze
 - vhodné pro aplikace zaměřené na analýzu velkého množství dat
 - využitelné i pro transakce
 - ale obtížnější
 - OLAP
 - online analytical processing
 - softwarové nástroje pro analýzu dat
 - platforma pro business analýzu (plánování, rozpočet, předpovědi, analýza) a následné rozhodování
 - velké množství dat

ORACLE

 riak



ONLINE ZPRACOVÁNÍ



<https://www.imaginarycloud.com/blog/oltp-vs-olap/>

VLASTNOSTI

- relační databáze
 - ACID vlastnosti
 - atomicita – atomicity
 - transakce se provede celá, nebo vůbec
 - konzistence – consistency
 - transakce nesmí narušit datovou integritu
 - izolovatelnost – isolation
 - současně probíhající transakce se nesmí ovlivnit
 - trvalost – durability
 - provedení transakce je zaznamenáno trvale
- NoSQL databáze
 - CAP teorém
 - pro distribuovaný datový sklad nelze zajistit více jak 2 následující vlastnosti
 - konzistence – consistency
 - všechny uzly mají stejná data
 - každé čtení vrací poslední výsledek, nebo chybu
 - dostupnost – availability
 - na každý dotaz je vrácena (nechybová) odpověď
 - odolnost k přerušení – partition tolerance
 - systém funguje dál i při zdržení či ztrátě zprávy
 - nezbytné pro big data
 - eventuálně konzistentní (AP)
 - často volena škálovatelnost a vysoká dostupnost než 100% konzistence
 - striktně konzistentní (AC)

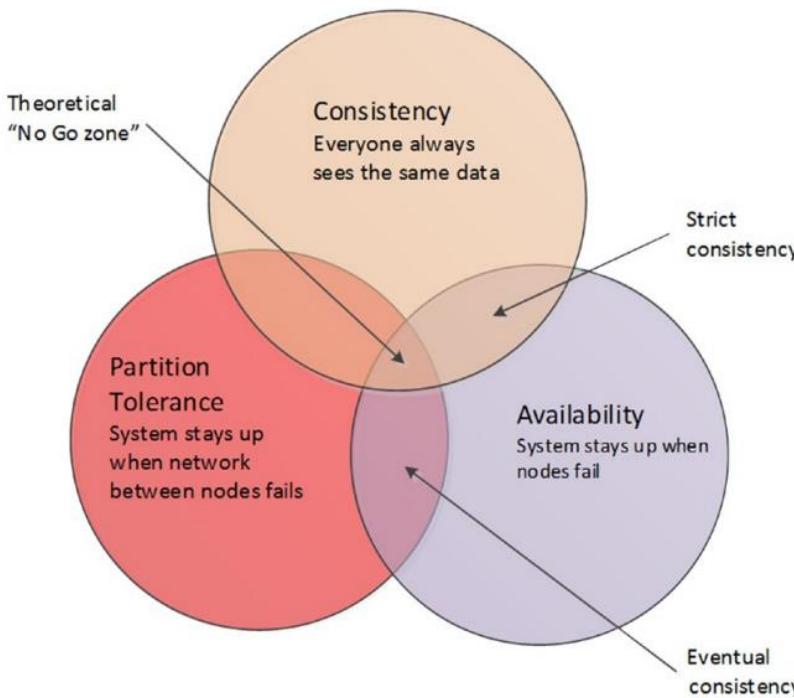


<https://www.tothenew.com/blog/overview-of-database-testing/>





CAP TEORÉM



HARRISON, Guy. *Next generation databases: NoSQL, NewSQL, and Big Data*. ISBN 978-1484213308.

DOTAZOVACÍ JAZYK

- relační databáze
 - specifický dotazovací jazyk
 - neliší se databázi od databáze
 - SQL
 - structured query language
 - příkazy rozděleny do skupin
 - DDL – data definition language
 - DML – data manipulation language
 - DCL – data control language
 - ostatní nebo speciální příkazy
- NoSQL databáze
 - nemá specifický jazyk na dotazy
 - liší se databázi od databáze
 - celá škála standardních jazyků
 - může podporovat i SQL



SLOŽITÉ DOTAZY

- relační databáze
 - jednodušší složité dotazy
 - pevně dané schéma
 - využití definovaných jmen relací a jednotlivých atributů
 - standardizovaný dotazovací jazyk
- NoSQL databáze
 - nemají standardní interface
 - nemají pevně daný jazyk
 - jazyk závislý na konkrétní databázi
 - limitované vazby
 - obtížnější složité dotazy



PODPORA

- relační databáze
 - dostupná podpora od všech prodejců
 - dlouhodobě zažitá a používaná technologie
 - velké množství odborníků
- NoSQL databáze
 - často open-source projekty
 - komunitní pomoc
 - omezenější počet odborníků
 - s rostoucí popularitou se situace postupně zlepšuje



SCÉNÁŘE

- relační databáze
 - účetní, finanční a bankovní systémy
 - systémy řízení zásob
 - systémy správy transakcí
- NoSQL databáze
 - mobilní aplikace
 - analýzy v reálném čase
 - správa obsahu
 - přizpůsobení
 - aplikace IoT
 - migrace databází





ZÁVĚREČNÉ SROVNÁNÍ

- výhody relačních databází
 - vztahy mezi daty – relace
 - velké množství vazeb a propojení
 - pevně dané schéma
 - komplexní dotazy
 - normalizace
 - databáze zabírá co nejméně místa
 - ideálně nulová redundance dat
 - známý dotazovací jazyk
 - datová integrita
 - smazání autora odstraní i jeho články
 - ACID vlastnosti
 - transakce se uskuteční celá, nebo vůbec





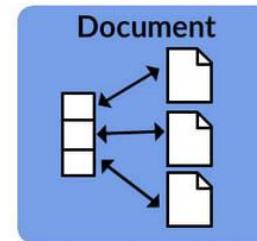
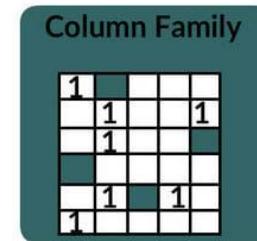
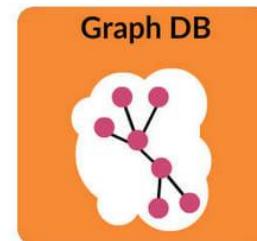
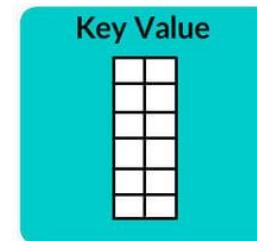
ZÁVĚREČNÉ SROVNÁNÍ

- výhody NoSQL
 - datový model
 - schéma není pevně definované
 - strukturovaná, částečně strukturovaná i nestrukturovaná data
 - flexibilní zpracování dat
 - možnost umístění v distribuovaných cloudech
 - horizontální škálování
 - replikace
 - práce s big data
 - analýza velkého množství dat
 - rychlosť, výkon
 - různé typy databází vyhovující různým aplikacím
 - levnější na údržbu



NoSQL – TYPY DATABÁZÍ PODROBNĚJI

- key-value úložiště
 - Riak, Redis, Couchbase Server
- dokumentové databáze
 - MongoDB, CouchDB, Elasticsearch
- sloupcová úložiště
 - Apache Cassandra, Druid, HBase
- grafová úložiště
 - Neo4j
- časové řady
 - InfluxDB



<https://dev.to/lmolivera/everything-you-need-to-know-about-nosql-databases-3o3h#key>



KEY-VALUE ÚLOŽIŠTĚ

- nejjednodušší typ NoSQL databází
- data ukládána pomocí klíče a přiřazené hodnoty
 - key-value
- podobné asociativnímu poli / hashi
- optimalizované pro velké objemy dat
 - ale jednoduchých (žádné vazby)
 - velmi rychlé
 - špatně upravitelné
 - dotazy vrací celou hodnotu
- např. ukládání uživatelských profilů
- Riak, Redis, Couchbase Server



DOKUMENTOVÉ DATABÁZE

- pravděpodobně nejpoužívanější NoSQL databáze
 - MongoDB
 - CouchDB, Elasticsearch
- data jsou ukládána v JSON / XML formátu
 - základem key-value páry
 - value je tzv. dokument
 - může obsahovat řetězce, čísla, pole, objekty, ...
- na úrovni databáze bez schématu
 - dynamické aplikace
 - např. uživatelé mohou přidávat ke svému profilu vlastní detailly
 - každý uživatel má jiné

```
{  
  "firstName": "John",  
  "lastName": "Smith",  
  "isAlive": true,  
  "age": 27,  
  "address": {  
    "streetAddress": "21 2nd Street",  
    "city": "New York",  
    "state": "NY",  
    "postalCode": "10021-3100"  
  },  
  "phoneNumbers": [  
    {  
      "type": "home",  
      "number": "212 555-1234"  
    },  
    {  
      "type": "office",  
      "number": "646 555-4567"  
    },  
    {  
      "type": "mobile",  
      "number": "123 456-7890"  
    }  
  "children": [],  
  "spouse": null  
}
```

<https://en.wikipedia.org/wiki/JSON>



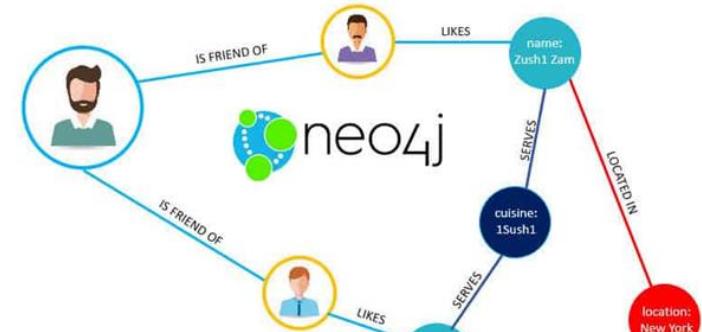
SLOUPCOVÁ ÚLOŽIŠTĚ

- optimalizovaná pro čtení a zápis ve sloupcích
 - místo řádků u relačních databází
- sloupcově orientovaný model pro ukládání dat
 - data ukládána do záznamů schopných pojmut v celém množství dynamických sloupců
 - klíče záznamů ani jména sloupců nejsou fixní
- minimalizují diskové I/O operace
- komprese dat
- vhodné pro analýzu dat
- Apache Cassandra, Druid, HBase



GRAFOVÁ ÚLOŽIŠTĚ

- grafový datový model
 - základem jsou vrcholy a hrany
 - hrany mohou být orientované
 - modeluje snadno vazby
 - žádné přerušené vazby
- použití pro velké objemy dat
 - s velkým množstvím propojení (vazeb)
 - sociální sítě
 - dopravní sítě...
- Neo4j



<https://www.bmc.com/blogs/neo4j-graph-database/>



DATABÁZE ČASOVÝCH ŘAD

- optimalizace pro časové řady
 - soubor pozorování získaný opakovaným měřením v čase
 - při nanesení na graf je jednou z os vždy čas
 - samotné body nenesou důležité informace, ty pochází z celé časové řady
 - data ze senzorů
 - využití CPU, MEM, ...
 - monitorování počasí
 - počet uživatelů v čase
- InfluxDB, Prometheus



Prometheus





ČÁST III.: KEY-VALUE ÚLOŽIŠTĚ

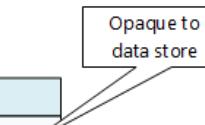


KEY-VALUE ÚLOŽIŠTĚ

- nejjednodušší typ NoSQL databází
- v podstatě velká zatřídovací tabulka
- data ukládána pomocí klíče (key) a přiřazené hodnoty (value)
 - každá hodnota přidružena k jedinečnému klíči
 - úložiště použije tento klíč k uložení dat pomocí odpovídající hash funkce
 - funkce je vybírána tak, aby zajistila rovnoměrné rozložení hashovaných klíčů
 - podobné asociativnímu poli / hashi

Key	Value
AAAAAA	1101001111010100110101111...
AABAB	100110000101100110101110...
DFA766	0000000000101010110101010...
FABCC4	1110110110101010100101101...

Opaque to data store



<https://docs.microsoft.com/cs-cz/azure/architecture/data-guide/big-data/non-relational-data>



KEY-VALUE ÚLOŽIŠTĚ

- většinou podpora jen základních operací
 - dotazování, vložení a odstranění
 - pokud je třeba hodnotu změnit (i jen částečně), musí být přepsána celá
 - operace čtení a zápis atomické
- hodnotou libovolná data
 - neprůhledná pro úložiště (blob)
 - úložiště je umí jen načíst a uložit podle klíče
- optimalizované pro jednoduché vyhledávání podle hodnoty klíče nebo rozsahu klíčů
 - méně vhodné pro dotazy napříč různými tabulkami klíčů a hodnot
 - spojování dat
 - neoptimalizované pro dotazování (filtrování) podle neklíčových hodnot



KEY-VALUE ÚLOŽIŠTĚ

- optimalizované pro velké objemy dat
 - ale jednoduchých (žádné vazby)
- extrémní škálovatelnost
 - snadná distribuce dat mezi uzly na samostatných počítačích
- významnější zástupci
 - Redis
 - Riak
 - Couchbase Server
 - Oracle NoSQL Database



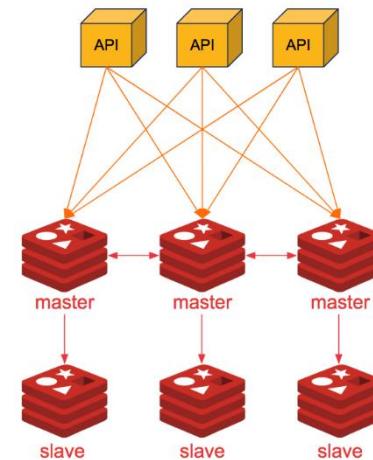
REDIS

- Remote Dictionary Server (od 2009)
 - distribuované paměťové open-source key-value úložiště
 - plní roli databáze, mezipaměti, zprostředkovatele zpráv a streamovacího enginu
 - v současné době nejpopulárnější zástupce key-value úložišť
 - úložiště je celé udržováno v paměti
 - extrémně rychlé čtení a zápisy s nízkou latencí
 - v základu zapisuje kopie dat do souborového systému každé dvě vteřiny
 - podporuje různé abstraktní datové typy
 - řetězce, listy, sety, seřazené sety, hashe
 - bitmapy, hyperlogy, proudy, geoprostorové indexy
 - replikace, shardování, klastrování
 - využívané celou řadou společností
 - Twitter, Airbnb, Tinder, Yahoo, Adobe, Hulu, Amazon, OpenAI, Github, Snapchat, ...



REDIS

- spousta vestavěných funkcí
 - asynchronní replikace
 - master-replica replikace
 - data replikována na libovolný počet replik
 - replika může být masterem jiné repliky
 - skriptování na straně serveru v Lua
 - volitelné úrovně perzistence na disku
 - podpora atomických operací a transakcí
 - klíče a hodnoty až do velikosti 512 MB
 - klíče jsou binárně bezpečné - od řetězce až po obrázek
 - využívá vlastní hashovací mechanismus - Redis Hashing
 - vysoká dostupnost přes Redis Sentinel
 - automatický partitioning přes Redis Cluster



<https://rtfm.co.ua/en/redis-replication-part-1-overview-replication-vs-sharding-sentinel-vs-cluster-redis-topology/>



REDIS

- **zabezpečení**
 - navržené pro přístup jen od důvěryhodných klientů
 - nevhodné pro externí přístup, vystavení na internetu
 - možnost nastavit jednoduchou autentizaci
 - chybí šifrování dat
- **hlavní výhody**
 - rychlosť (paměť)
 - až 110000 / 81000 SET / GET příkazů za sekundu
 - kešování
 - flexibilní datové struktury
 - jednoduchost a uchopitelnost
 - replikace a škálovatelnost

- An empty instance uses ~ 3MB of memory.
- 1 Million small Keys -> String Value pairs use ~ 85MB of memory.
- 1 Million Keys -> Hash value, representing an object with 5 fields, use ~ 160 MB of memory.

<https://redis.io/docs/get-started/faq/>

ActionScript	Bash	C	C#	C++	Clojure
Common Lisp	Crystal	D	Dart	Delphi	Elixir
emacs lisp	Erlang	Fancy	gawk	GNU Prolog	Go
Haskell	Haxe	Io	Java	Julia	Lasso
Lua	Matlab	mruby	Nim	Node.js	Objective-C
OCaml	Pascal	Perl	PHP	Pure Data	Python
R	Racket	Rebol	Ruby	Rust	Scala
Scheme	Smalltalk	Swift	Tcl	VB	VCL

<https://www.youtube.com/watch?v=Hbt56gFj998>



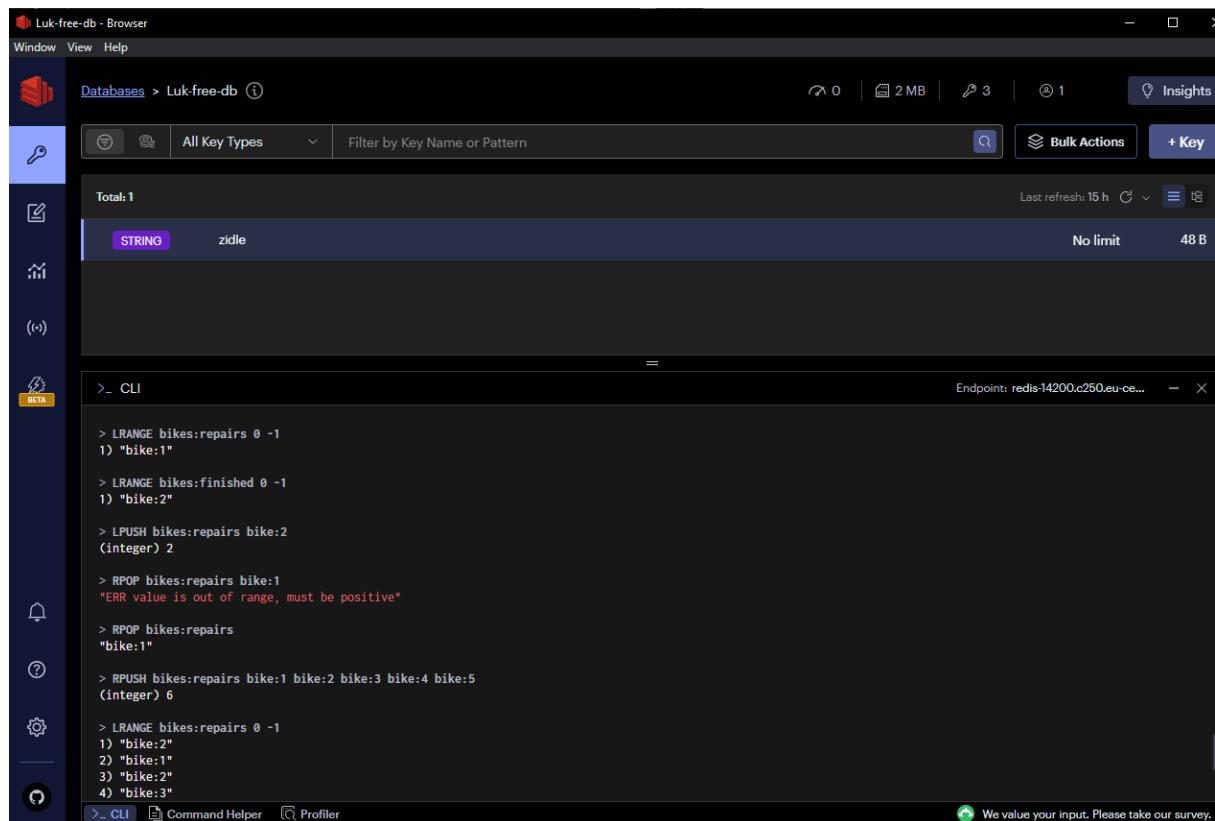
REDIS STACK

- balíček technologií rozšiřující funkcionalitu klasického Redis
 - kompletní vývojářské prostředí pro ladění a vývoj
 - pravděpodobnostní datové struktury
 - dotazovatelné JSON dokumenty
 - dokumentová databáze
 - dotazování napříč hash a JSON dokumenty
 - podpora časových řad včetně fulltextového vyhledávání
 - vizualizační nástroj RedisInsight



REDIS STACK

- RedisInsight



REDIS

- stažení & instalace



Redis

You can download the last Redis source files here. For additional options, see the [Redis downloads](#) section below.

Stable (7.2)

Redis 7.2 includes optimizations, several new commands, some improvements, bug fixes, and several new module APIs. It also includes changes that potentially break backwards compatibility with older versions.



Redis Stack

Download the latest Redis Stack Server binaries here, or install with [Docker](#), [Homebrew](#), or [on Linux](#).

Stable (7.2)

Redis Stack Server extends Redis with modern data models such as document, time series. Redis Stack also includes RedisInsight, a visualization tool for Redis. Read the [latest release notes](#), or download the latest 7.2 binaries:

- možnost cloudu

REDIS

- Docker Hub
 - docker run --name *jméno* -d redis
 - docker container exec -it *jméno* bash
- Redis prakticky
 - redis-cli
 - ověření připojení
 - PING
 - ECHO 'zpráva'
 - ukončení připojení
 - QUIT
 - snapshot na disk
 - SAVE

```
root@3fc37931dcc4:/# redis-cli
127.0.0.1:6379> ping
PONG
```

```
> ECHO "ja jsem tucnak"
"ja jsem tucnak"
```



REDIS PRAKTICKY

- základní datový typ - řetězec
 - ukládají sekvence bytů
 - včetně textů, serializovatelných objektů a binárních polí
 - reprezentují Redis klíče
 - nejjednodušší typ přiřaditelný jako hodnota
 - vložení dat SET klíč hodnota
 - čtení dat GET klíč
 - přepsání a vrácení původní hodnoty GETSET klíč
 - vícenásobné vložení hodnoty MSET klíč hodnota, hodnota, ...
 - vícenásobné přečtení hodnot MGET klíč, klíč, ...
 - inkrementace / dekrementace hodnoty INCR / DECR klíč
 - o více pomocí INCRBY / DECRBY

```
127.0.0.1:6379> SET foo 100
OK
127.0.0.1:6379> SET bar "HELLO WORLD"
OK
```

```
127.0.0.1:6379> GET foo
"100"
```

```
127.0.0.1:6379> INCR foo
(integer) 101
127.0.0.1:6379> DECR foo
(integer) 100
```

```
127.0.0.1:6379> MSET key1 "Hello" key2 "world"
OK
```



REDIS PRAKTICKY

- další operace
 - seznam existujících klíčů přes SCAN
 - přejmenování klíče RENAME klíč nový_klíč
 - kontrola existence klíče EXISTS klíč
 - smazání dat DEL klíč
 - vymazání všech klíčů FLUSHALL
 - vypršení dat EXPIRE klíč čas
 - zbývající čas přes TTL klíč
 - alternativně SETEX klíč čas hodnota
 - PERSIST zastaví vypršení
 - přidání k hodnotě pomocí APPEND klíč hodnota
 - délka hodnoty přes STRLEN klíč

```
SCAN 0 MATCH "bike:/" COUNT 100
```

```
127.0.0.1:6379> SET greeting "Hi"
OK
127.0.0.1:6379> EXPIRE greeting 50
(integer) 1
127.0.0.1:6379> TTL greeting
(integer) 44
127.0.0.1:6379> GET greeting
"Hi"
127.0.0.1:6379> GET greeting
"Hi"
127.0.0.1:6379> TTL greeting
(integer) -2
127.0.0.1:6379> GET greeting
(nil)
```



REDIS PRAKTICKY

- další datové typy - list
 - list řetězců seřazený podle data vložení
 - implementované jako spojový seznam
 - rychlé vkládání na začátek / konec seznamu
 - ale pomalejší přístup k prvkům podle konkrétního indexu
 - vkládání pomocí LPUSH / RPUSH list hodnota
 - čtení pomocí LRANGE list index_start index_konec
 - počet elementů LLEN list
 - LPOP a RPOP vrátí prvek a zároveň ho odstraní z listu
 - LTRIM pro omezení velikosti listu
 - možnost snadno implementovat frontu / zásobník

```
> LPUSH bikes:repairs bike:1
(integer) 1

> LPUSH bikes:repairs bike:2
(integer) 2

> RPOP bikes:repairs
"bike:1"

> RPOP bikes:repairs
"bike:2"
```





REDIS PRAKTICKY

- další datové typy - set

- nesetříděná kolekce unikátních řetězců
- vložení SADD set hodnota
- SISMEMBER pro kontrolu, jestli člen existuje
- SMEMBERS pro výpis
 - SRANDMEMBER pro náhodný prvek
- SCARD pro počet prvků
- SREM pro odstranění člena
- SINTER pro průnik n setů

```
127.0.0.1:6379> SADD auta "Ford" "Skoda" "BMW"
(integer) 3
127.0.0.1:6379> SMEMBERS auta
1) "BMW"
2) "Skoda"
3) "Ford"
127.0.0.1:6379> SCARD auta
(integer) 3
```

```
> SADD bikes:racing:france bike:1 bike:2 bike:3
(integer) 3
> SADD bikes:racing:usa bike:1 bike:4
(integer) 2
> SADD bikes:racing:italy bike:1 bike:2 bike:3 bike:4
(integer) 4
> SINTER bikes:racing:france bikes:racing:usa bikes:racing:italy
1) "bike:1"
> SUNION bikes:racing:france bikes:racing:usa bikes:racing:italy
1) "bike:2"
2) "bike:1"
3) "bike:4"
4) "bike:3"
> SDIFF bikes:racing:france bikes:racing:usa bikes:racing:italy
(empty array)
> SDIFF bikes:racing:france bikes:racing:usa
1) "bike:3"
2) "bike:2"
> SDIFF bikes:racing:usa bikes:racing:france
1) "bike:4"
```





REDIS PRAKTICKY

- další datové typy - hash

- typ záznamů strukturovaných jako kolekce párů klíč - hodnota
 - vhodné pro reprezentaci objektů
- vložení pomocí HSET, HMSET
- čtení přes HGET, HMGET, HGETALL
- HKEYS, HVALS
- HDEL

```
127.0.0.1:6379> HSET user:luke name "Luke"
(integer) 1
127.0.0.1:6379> HSET user:luke email "mail@gmail.com"
(integer) 1
127.0.0.1:6379> HGET user:luke name
"Luke"
127.0.0.1:6379> HGET user:luke email
"mail@gmail.com"
```

```
127.0.0.1:6379> HMSET user:jane name "Jane" age "25" email "m@gmail.com"
OK
127.0.0.1:6379> HGETALL user:jane
1) "name"
2) "Jane"
3) "age"
4) "25"
5) "email"
6) "m@gmail.com"
127.0.0.1:6379> HKEYS user:jane
1) "name"
2) "age"
3) "email"
```

```
127.0.0.1:6379> HGETALL user:luke
1) "name"
2) "Luke"
3) "email"
4) "mail@gmail.com"
```



REDIS PRAKTICKY

- další datové typy - setříděný set
 - kolekce unikátních řetězců setříděných podle přiřazeného skóre
 - v případě stejného skóre lexikografické řazení
 - leží na pomezí setů a hashů
 - vložení ZADD set skóre hodnota
 - ZRANK set hodnota vrací index po řazení
 - ZRANGE, ZREVRANGE pro výpis
 - WITHSCORES vrátí i skóre
 - ZRANGEBYSCORE, ZREVRANGEBYSCORE
 - ZCARD, ZCOUNT

```
>ZRANGE racer_scores 0 -1 withscores
1) "Ford"
2) "6"
3) "Sam-Bodden"
4) "8"
5) "Norem"
6) "10"
7) "Royce"
8) "10"
9) "Castilla"
10) "12"
11) "Prickett"
12) "14"
```

```
127.0.0.1:6379> ZADD users 1981 "Brad"
(integer) 1
127.0.0.1:6379> ZADD users 1990 "Jen"
(integer) 1
127.0.0.1:6379> ZADD users 1991 "Luke"
(integer) 1
```



REDIS PRAKTICKY

- transakce
 - skupina příkazů vykonávaných v jediném kroku
 - spuštěné sekvenčně jako izolovaná operace
 - transakce začíná příkazem MULTI
 - následují požadované příkazy
 - vkládané do fronty a čekající na vykonání
 - transakce je provedena příkazem EXEC
 - příkazem DISCARD je možné transakci přerušit
 - WATCH pro sledování klíče
 - transakce je provedena, jen pokud nedošlo ke změně klíče
 - v případě selhání určitého příkazu jsou všechny ostatní provedeny
 - rollback není podporován kvůli jednoduchosti a výkonu Redisu

```
> MULTI
OK
> INCR foo
QUEUED
> INCR bar
QUEUED
> EXEC
1) (integer) 1
2) (integer) 1
```



VÝZNAMNĚJŠÍ ZÁSTUPCI

- Riak (od 2009)
 - distribuované key-value úložiště
 - licence Apache 2
 - hlavní vlastnosti
 - vysoká dostupnost
 - volba konzistence
 - eventuální nebo silná (CAP)
 - odolnost vůči chybám
 - data mohou být ukládána do paměti, na disk, případně kombinovaně
 - škálování
 - replikace
 - Riak využívá např.
 - AT&T, Comcast, GitHub, Riot Games, The Weather Channel...



VÝZNAMNĚJŠÍ ZÁSTUPCI

- Couchbase Server (od 2010)
 - původně Membase
 - open-source cloudová databáze
 - více modelová
 - key-value přístup
 - dokumentový přístup
 - JSON
 - ale i jiné formáty jsou podporovány
- Couchbase Mobile
 - pro mobilní a embedded zařízení



VÝZNAMNĚJŠÍ ZÁSTUPCI

- Oracle NoSQL Database (od 2011)
 - distribuované key-value úložiště od Oracle Corporation
 - licence Apache 2
 - postavené na Oracle Berkeley DB
 - podporuje
 - sharding
 - horizontální škálování
 - replikaci
 - ACID transakce
 - jednoduchou administraci a monitoring



ONLINE KURZY

- Redis

<https://www.udemy.com/course/learn-redis-from-scratch/>

- MongoDB

<https://www.coursera.org/learn/introduction-mongodb>

<https://www.udemy.com/course/mongodb-the-complete-developers-guide/>

<https://www.udacity.com/course/data-wrangling-with-mongodb--ud032>

- Elasticsearch

<https://www.udemy.com/course/elasticsearch-complete-guide/>

- Apache Cassandra

<https://www.udemy.com/course/apache-cassandra/>

- Neo4j

<https://www.udemy.com/course/neo4j-foundations/>



LITERATURA

- HARRISON, Guy. *Next generation databases: NoSQL, NewSQL, and Big Data*. ISBN 978-1484213308.
- SULLIVAN, Dean. *NoSQL for Mere Mortals*. ISBN 978-0134023212.
- TIWARI, Shashank. *Professional NoSQL*. ISBN 978-0470942246.



A PŘÍŠTĚ?

- dokumentové databáze





Děkuji za pozornost.
Otázky?

