Úvod do NoSQL databáz

EURÓPSKA ÚNIA
Európsky sociálny fond
Európsky fond regionálneho rozvoja

OPERAČNÝ PROGRAM
ĽUDSKÉ ZDROJE

MINISTERSTVO ŠKOLSTVA, VEDY, VÝSKUMU A ŠPORTU SLOVENSKEJ REPUBLIKY

Tento projekt sa realizuje vďaka podpore z Európskeho sociálneho fondu a Európskeho fondu regionálneho rozvoja v rámci Operačného programu Ľudské zdroje

www.minedu.sk www.employment.gov.sk/sk/esf/ www.itakademia.sk

Big Data

- Dnešný trend:
  - veľa používateľov (vyše 2 miliardy je online)
  - všetko je na cloud-e
- Ak človek postaví webovú službu, je bežné, že za pár mesiacov má milióny používateľov
- „3 V“:
  - Volume (terabajty až zetabajty)
  - Velocity(prúdy dát)
  - Variety (rôzne štruktúrované až neštruktúrované)
  - Veracity (neurčitosť, nekonzitentnosť, nekompletnosť, nejednoznačnosť)
- Bonusové V:
  - Value (biznisová hodnota dát)
  - Validity (korektnosť a presnosť dát)
  - Volatility (obmedzená platnosť iba v danom čase)

Práca s veľkými dátami

- Nepresný formát dát
- Málokedy sa deje zmena dát
- Dáta majú obmedzenú platnosť a potom prichádzajú novšie
- Počet čítaní prevažuje nad počtom zápisov (write-once/read-many)
- Počet dát môže veľmi rýchlo, až exponenciálne rásť
- Základný nástroj: distribuované spracovanie
  - namiesto silnejšieho počítača veľa počítačov

Technológie pre veľké dáta

- Distribuované súborové systémy (GoogleFS, HDFS,..)
- MapReduce
- Spracovanie paralelných prúdov dát (Apache Beam nad Apex, Flink, Spark, …)
- NoSQL databázy
- Dátové sklady
- Gridové a klaudové počítanie
- Strojové učenie nad veľkými dátami

Relačné databázy

- Dominantná forma databázy, používaná na rôzne účely
- Pre niektoré problémy to nemusí byť najlepší spôsob narábania s dátami
- Iné modely:
  - Hierachický
  - Objektovo – orientovaný
  - XML
  - Grafový
  - Úložiská typu kľúč-hodnota
  - Dokumentové modely

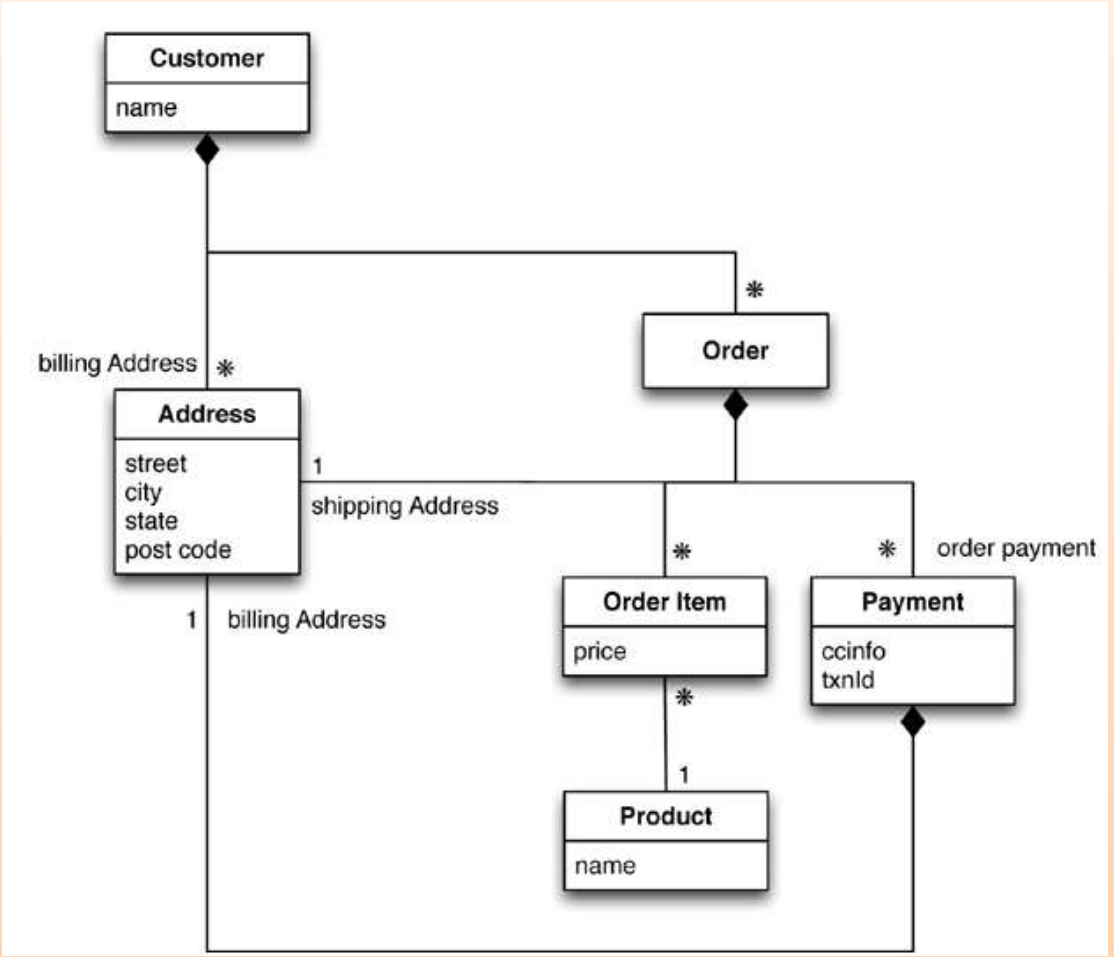
Obmedzenia v relačných databázach

Niektoré vlastnosti relačných databáz môžu byť v niektorých prípadoch obmedzujúce

1. Pevná schéma
   - schéma musí byť pripravená pred vložením dát, zmeny schémy sú náročné
   - mnohé reálne dáta nie sú konzistentné
2. Náročné spájanie dát
   - tabuľky v BCNF alebo v 3. NF spôsobia rozbitie dát do mnohých tabuliek a ich opätovné vyskladanie si vyžaduje veľa spájaní – pomalé hlavne v distribuovanom prostredí
3. Striktné transakcie
   - nie každá zmena musí zamykať tabuľky na čítanie, ak mi nezáleží na aktuálnosti
4. Škálovanie
   - architektúra štandardnej databázy predpokladá uloženie na jednom stroji
   - paralelné databázy zamerané na konzistentnosť dát výrazne obmedzujú možnosti práce s veľkými dátovými tokmi

JSON vs. relačná DB

```
// in customers
{
"customer": {
"id": 1,
"name": "Martin",
"billingAddress": [{"city": "Chicago"}],
"orders": [
  {
    "id":99,
    "customerId":1,
    "orderItems":[
    {
    "productId":27,
    "price": 32.45,
    "productName": "NoSQL Distilled"
    }
  ],
  "shippingAddress":[{"city":"Chicago"}]
  "orderPayment":[
    {
    "ccinfo":"1000-1000-1000-1000",
    "txnId":"abelif879rft",
    "billingAddress": {"city": "Chicago"}
    }],
  }]
}
}
```



Horizontálne škálovanie

- Vertikálne škálovanie = zvyšovanie výpočtového výkonu servera
- Horizontálne škálovanie = zvyšovanie počtu výpočtových uzlov
- Problémy HŠ, s ktorými treba rátať
  - Nespoľahlivosť siete
  - Latencia
  - Obmedzená prenosová rýchlosť
  - Bezpečnosť sieťovej komunikácie
  - Možná zmena topológie siete
  - Viacero správcov sietí a počítačov
  - Nehomogénne konfigurácie sietí a počítačov

Relačné databázy a ACID vlastnosti

Not Only SQL

- ACID vlastnosti
  - Atomicity – buď sa vykoná celá transakcia alebo nič
  - Consistency – transakcie menia databázu z jedného konzistetného stavu na iný konzistentný stav
  - Isolation – nekomitnuté zmeny v jednej transakcii sú pre druhú transakciu neviditeľné
  - Durability – zmeny vykonané komitnutými transakciami pretrvajú (aj keby hneď vypadla elektrina)
- Umožňujú konkurentný prístup so zachovaním konzistencie a aktuálnosti dát

CAP tvrdenie

- Autor Eric Brewer
- Tvrdí, že systém na správu dát môže mať iba 2 z nasledujúcich 3 vlastností
  - Konzistentnosť (Consistency)
    - Po zmene dát vidia všetci klienti tie isté dáta
  - Dostupnosť (Availability)
    - Všetky čítania a zápisy sú úspešné aj, keď niektorý server padol
  - Odolnosť voči rozpadnutiu siete (Partition toleance)
    - Ak sa množina serverov rozpadne na viacero, ktoré sa navzájom nevidia, vzniknuté ostrovy serverov sú funkčné
- Tradičná DB nie je odolná voči rozpadu siete (C-A)
- Väčšina NoSQL sytémov obetuje trochu konzistentnosti, aby zostala odolná a väčšinou dostupná (¾C- ¾A - P – čísla bez opory o čokoľvek)

Občasná konzistencia

- Zmeny spôsobia dočasnú nekonzistentnosť, ktorá po čase zanikne, keď sa zmeny rozšíria do celého systému
  - Niektorí klienti môžu prečítať staršie dáta
- BASE = Basically Available, Soft state, Eventual consistency
  - Prevažne dostupný systém, niekedy dochádza k výpadkom časti systému
  - Systém je dynamický, stále dochádza k zmenám

Distribúcia dát

- Dve nezávislé techniky distribúcie dát
  - Replikácia – rovnaké dáta sú na viacerých serveroch
  - Delenie dát – kolekcie dát nie sú uložené v jednom kuse, ale po častiach na viacerých serveroch
    - Horizontálne delenie – dáta o jednom objekte sú inde ako o inom
    - Vertikálne delenie – niektoré informácie o objekte sú inde ako iné informácie o tom istom objekte

Master-slave replikácia

- Jedna kópia je master kópia
- Ak meníme dáta, musíme meniť master kópiu, odkiaľ sa nová hodnota rozdistribuuje
- Čítame zo všetkých kópií
- Ak master padne, ostatné kópie sa naďalej dajú čítať
  - V závislosti od filozofie databázy sa môže nová master kópia zvoliť zo zvyšných kópií
- Vhodné, ak sa oveľa viac číta ako zapisuje
  - Master je úzke miesto
- Uzol môže byť master pre nejaké dáta a slave pre iné

Peer-to-peer replikácia

- Všetky kópie sú si rovné
- Hrozí write-write konflikt
  - Riešenie je možné pomocou väčšiny:
    - Ak zapisujem, musím zapísať do nadpolovičnej množiny kópií: W > N/2
    - Ak si chcem byť istý, že prečítam aktuálne dáta, stačí že prečítam polovicu kópií (dáta musia mať časovú pečiatku): R > N – W
    - N nezvykne byť veľké, obvykle je N = 3

NoSQL databázy

Not Only SQL

- Použiteľné pre mnohé problémy, kde je štandardná DB skôr obmedzením
  - Variabliné dáta: slabo štruktúrované, meniace sa v čase, alebo úplne ľubovoľné, ale organizované do skupín „podobných dát“
  - Obrovské dáta: veľa terabajtov až petabajty
  - Vysoký paralelizmus: veľa používateľov, veľa počítačov, jednoduchá škálovateľnosť, odolnosť na zlyhanie HW
  - Jednoduché dopyty: nepotrebujeme plnú silu SQL
  - Slabá konzistencia: nevieme či dostaneme najnovšie dáta, zmeny sa prejavia až po čase
  - Lacné: predpoklad použitia mnohých, ale pokojne aj lacných bežných počítačov
  - Netreba robiť zálohy – replikácia a zotavenie z chýb je súčasťou systémov
- NoSQL = Not Only SQL
  - Znamená: SQL nie je jediné správne riešenie na všetko
  - Nie je to o SQL, ale o nových spôsoboch správy dát

NoSQL – slabé stránky

- Relatívne mladé - mnoho vecí je ešte nevyladených, nedoprogramovaných
- Slabá podpora – väčšinou open source
- Náročná administrácia – väčšia námaha pri inštalácii a správe systému
- Zložitá analytika – iba jednoduché operácie, zložitejšie vyhodnotenia vyžadujú programovanie
- Málo expertov
- Nejednotné API – relačné DB majú podobné konektory a spoločný jazyk SQL, NoSQL majú rozdielne prístupy (API) aj v rámci spoločného typu

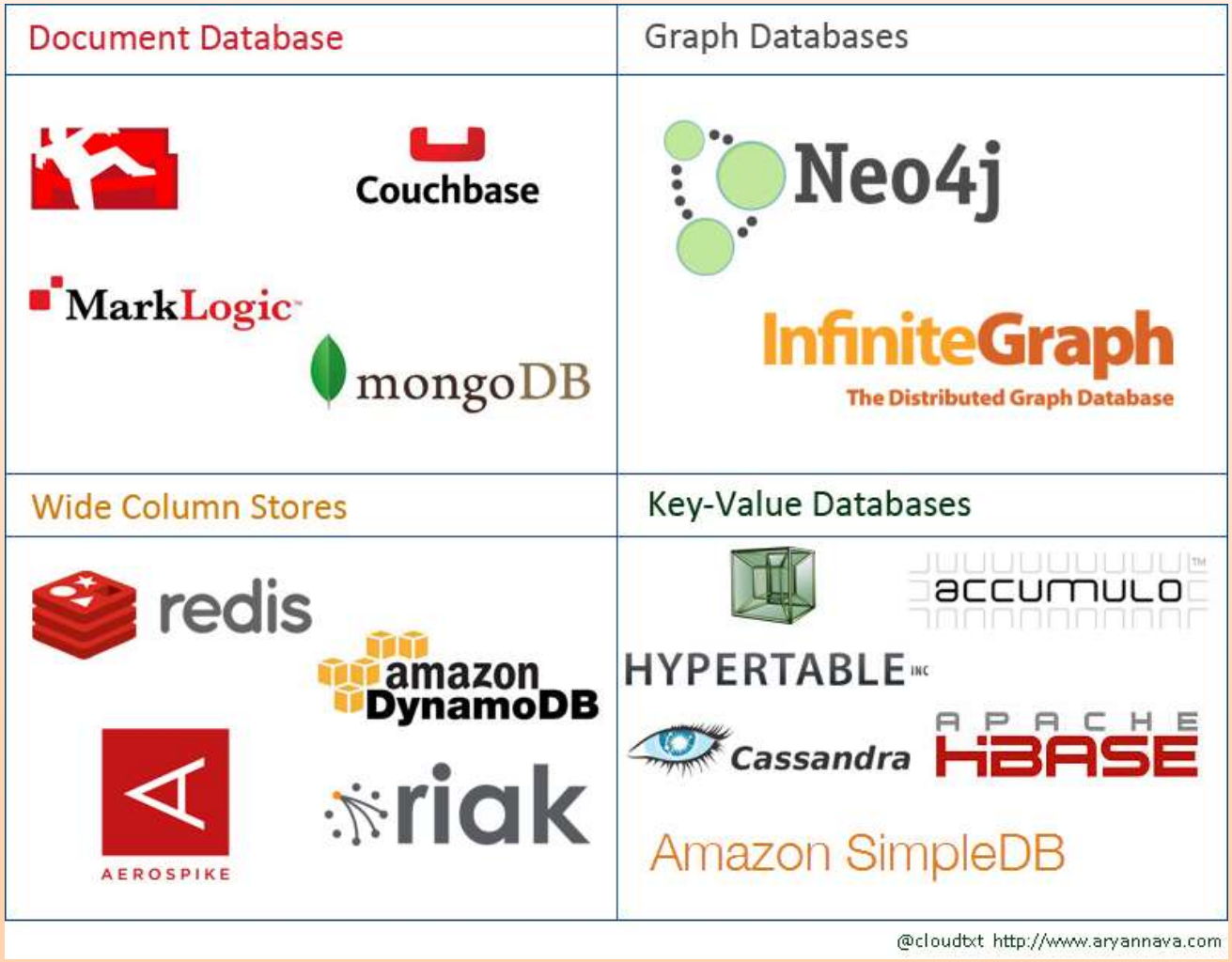
Typy NoSQL

- Kľúč-hodnota
  - Je to v podstate mapa uložená na disku/diskoch
- Stĺpcové DB
  - Najpodobnejšie relačným tabuľkám
  - Počet stĺpcov nie je pre každý riadok fixný
- Dokumentové DB
  - Ukladajú neštruktúrované alebo pološtruktúrované texty napr. JSON
- Grafové DB
  - Ukladajú grafy uzlov a hrán

Ústav informatiky
Prírodovedecká fakulta
UPJŠ v Košiciach

Not Only SQL

Každý typ má mnoho zástupcov



Not Only SQL

Čo NoSQL obvykle nemajú

- Spojenia (join)
- Group by
- Order by
- ACID vlastnosti a transakcie
- Deklaratívne dopyty
- JDBC/ODBC
- Podobný jazyk medzi rôznymi systémami rovnakého typu (SQL)

Úložiská typu kľúč-hodnota

- Ukladajú a poskytujú dáta na základe kľúčov
  - Ako hash mapa len distribuovaná – často Distributed Hash Table
- Hodnoty sú hociaké dáta (niektoré DB uvažujú viac typov dát ako len hocičo)
- Optimalizované na rýchlosť (pamäťová implementácia)
- Operácie
  - Insert(kľúč, hodnota)
  - Fetch(kľúč)
  - Update(kľúč, hodnota)
  - Delete(kľúč)
- Dá sa to predstaviť ako dvojstĺpcová relačná tabuľka s ID a BLOB stĺpcami
- Typické použitie: uloženie webového sedenia podľa session ID, zdieľané aktívne štruktúry medzi procesmi v distribuovanom prostredí, ...
- Kedy ich nepoužiť: modelovanie vzťahov medzi objektmi, hľadanie podľa dát, získavanie množiny záznamov / intervaly kľúčov, zložité dáta

Úložiská typu kľúč-hodnota

- Výhody: vysoko škálovateľné, odolné voči výpadkom, veľmi rýchle
- Obmedzenia: zamykanie po jednotlivých záznamoch, občasná konzistencia, primitívne API s minimom funkcionalít, nemožnosť vyhľadávať nad hodnotami
- Systémy: Redis (in-memory), Memcached(in-memory), Hazelcast(in-memory v Jave), Ehcache(in-memory v Jave), Riak KV (na disku), ...
  - Viď: https://db-engines.com/en/ranking/key-value+store

Stĺpcové DB

- Ukladajú dáta ako záznamy s potenciálne veľkým počtom dynamických stĺpcov bez vopred danej schémy
- Ukladajú dáta nie po riadkoch, ale po stĺpcoch, resp. rodinách stĺpcov
- Niečo ako dvojrozmerný variant DB typu kľúč-hodnota (lebo kľúč-rodina stĺpcov-hodnoty v stĺpcoch)
- Typické použitie: logovanie udalostí – vyplnia sa len relevantné stĺpce, CMS (content management systems) – voliteľné tagy, kategórie a ich atribúty, odkazy, komentáre, hodnotenia
- Kedy ich nepoužiť: ak vyžadujeme transakcie a konzistentné dáta, potreba agregácií riadkov alebo zložitejších dopytov na základe hodnôt v rôznych stĺpcoch – to sa dá len ďalším programovaním

Stĺpcové DB

- Výhody: najpodobnejšie relačným tabuľkám, veľmi rýchle, vysoko škálovateľné, odolné voči výpadkom, ukladajú viacero verzií dát
- Obmedzenia: občasná konzistencia, obmedzenie funkcionality oproti dopytom v relačných DB
- Systémy: Cassandra, Hbase, BigTable,...

Ústav informatiky
Prírodovedecká fakulta
UPJŠ v Košiciach

Dokumentové DB

- Optimalizované pre textové pološtruktúrované dokumenty – typicky JSON alebo XML
  - Hierarchická štruktúra dát
  - Objekty v jednej kolekcii by mali byť štrukturálne podobné
- Dokumenty sa dajú vyhľadať podľa kľúča, ale aj podľa obsahu dokumentu
- Typické použitie: logovanie udalostí, CMS, blogy, e-shopy, dokumenty, ...
- Kedy ich nepoužiť: množinové operácie cez viacero dokumentov, príliš rozdielna schéma dokumentov

Ústav informatiky
Prírodovedecká fakulta
UPJŠ v Košiciach

Dokumentové DB

- Výhody: vysoko škálovateľné, odolné voči výpadkom, rýchle najmä pri využití indexov
- Obmedzenia: môže spôsobiť redundanciu, rôzne obmedzenia pri vyhľadávaní nad dokumentmi, bez použitia indexov môžu byť pomalé
- Systémy pre JSON: MongoDB, Amazon DynamoDB, Couchbase, CouchDB, ElasticSearch...
- Natívne XML systémy: MarkLogic, BaseX,...
  - Dopytovanie cez XPath, Xquery a XSLT

Not Only SQL

Grafové DB

- Modelujú dáta ako vrcholy a hrany
  - Vrcholy majú atribúty, ako objekty s primitívnymi inštančnými premennými
  - Hrany majú smer a pomenovanie (modelujú vzťahy medzi objektmi – zložitejšie inštančné premenné)
- Typické použitie: sociálne siete, geograficky-založené dáta, chemické biologické zlúčeniny, znalostné dáta, rôzne vzťahy medzi entitami mnohých typov, modelovanie procesov
- Kedy ich nepoužiť: práca s veľkým množstvom uzlov naraz, ukladanie obrovských grafov vyžadujúce distribúciu dát

Grafové DB

- Výhody: optimalizované na prechod grafom, čo štandardné DB realizujú drahými JOIN operáciami
- Obmedzenia: nie pre veľmi veľké dáta, slabá škálovateľnosť vyplývajúca z povahy grafovej štruktúry
- Všeobecné: Neo4J, Titan, Apache Giraph, ...
- RDF/OWL databázy: Jena, RDF4J, RedStore, ...
  - Dopytovanie cez SPARQL

Objektové DB

- Vývoj už od 80-tych rokov
- Mnohé súčasné relačné DB boli rozšírené o OO črty, ako sú používateľom definované typy a štruktúrované atribúty
- Posunuté do úzadia s príchodom ORM (objektovo-relačné mapovanie) systémov ako Hybernate alebo JPA
- Systémy: Caché, Db4o, Versant,...

Otázky?