Kardkovács Zsolt Tivadar, Kerepes Tamás

Adatbázisok

NoSQL adatbázisok

„[cloud computing] a hirdetéseinkben a szóhasználat megváltozása.”

(Larry Ellison)

https://www.youtube.com/watch?v=0FacYAI6DY0

Melyek a mai relációs adatbázisok értékei?

- Egyszerű alapelvek
- Matematikai alapok
- Szabványos nyelv: SQL
- ACID (Atomicity, Consistency, Isolation, Durability
- Hatalmas felhasználói bázis
- Rengeteg tapasztalat és siker-sztori
- Bizonyítottak 40 év alatt milliónyi incidens-helyzetben
- Sok-sok esetleges extra „szolgáltatás”
- …

Mi a gond a relációs adatbázisokkal?

- Az adatok mennyisége elképesztően növekszik
- A struktúrált üzleti adatok túlnyomó többségét ma is relációsan tároljuk
- A mennyiség miatt esetleg nem tudjuk betartani a hatékonysági elvárásokat (lelassulunk)
- Az előre nem definiált ad-hoc szerkezetű adatok nem „ízlenek” a relációs modellnek és a relációs adatbáziskezelőknek

- A megoldási javaslat: valamiféle skálázódás. Nagyobb gép („Scale Up”), vagy több gép („Scale Out”).
- Mivel a hardver teljesítménye lemarad az adatok mennyiségéhet képest, a Scale Out tűnik járhatónak

A lehetséges megoldás: elosztott adatbázisok, elosztott feldolgozás

Elsődleges mozgatóerők
- Teljesítménynövekedés
  - Időegység alatt több tranzakció
  - vagy gyorsabb lefutás

Distributed, Parallel, Clusters, Grids, Internet → Cloud

Észrevétel
- Párhuzamosíthatóság
  - Relációs logikai műveleteké
  - Tranzakció-kezelésé

Ötlet
- Osszuk szét a feladatokat…

Növekedő vállalat = kinőtt szerverek

Dilemmák
- Mit osszunk meg?
- Több (scale out) vagy nagyobb gépek (scale up)?
- Áteresztőképesség vagy válaszidő?

Az elosztott feldolgozás előnyei

- Megbízhatóság (Reliability)
- Skálázhatóság (Scalability)
- Az erőforrások megosztása (Resource sharing)
- Flexibilitás
- Sebesség (Speed)
- Ezek többségükben nyílt rendszerek

Az elosztott feldolgozás hátrányai

- Nehezebb a hibakeresés
- Gyengébb szoftver-támogatás
- Potenciális hálózati problémák
- Biztonság (a nagy elosztottság, a sok rendszer miatt)

Big Data

Mi van, ha tényleg nagy adattal akarunk dolgozni?

- Big Data jellemző (IBM alapján)
  - (**Volume**) „végtelen” nagy – azaz nagy méretű
  - (**Variety**) változatos – azaz struktúrában gazdag adathalmaz
  - (**Velocity**) valósidejű – azaz azonnal kell
- Kritikus (nem megcélzott!) pontok
  - (**Veracity**) valós – azaz a megkapott információ (elég) konzisztens
  - (**Verifiability**) validálható – azaz eredmény ellenőrizhető
- Hol merül fel?
  - Internetszolgáltatások
  - „Internet of Things”
  - Nagyvállalati összekapcsolt rendszerekben, pl.
    - Számlázási rendszer, adattárházak
    - Ügyfélkezelés, hibakezelés, üzemeltetés
    - Kapcsolati hálók (HR, PR, xR)

Eric Brewer sejtése

Alapgondolat
(http://www.cs.berkeley.edu/~brewer/cs262b-2004/PODC-keynote.pdf)

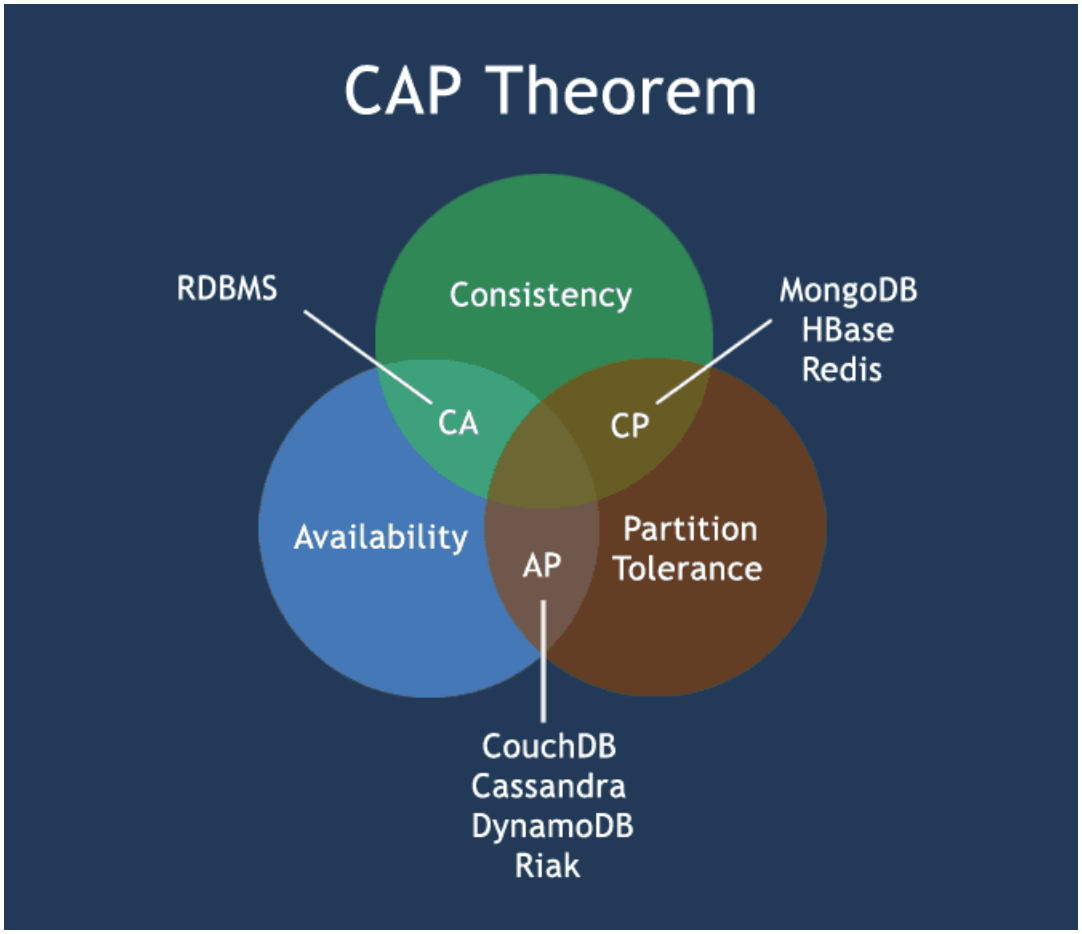
- Elosztott rendszerek nem működnek web méretekben, mert…
  - …a folyamatokra fókuszálnak az adatok helyett(!)
  - …de a probléma az adatok megosztásában/frissítésében van

- Elosztott (web)rendszer esetében
  - (C) elosztott / globális konzisztencia(!)
  - (A) hozzáférés + rendelkezésre állás
  - (P) partíciók hibatűrése is kellene

- „CAP-sejtés” (2000) vagy „CAP-tétel” (2002)
  - „A konzisztencia, a rendelkezésre állás és a partíciók hibatűrése esetében a háromból egyszerre legfeljebb kettő garantálható.”

Értsd:„trade-off”…

A CAP tétel és egyes adatbáziskezelők viszonya



Forrás: w3resource „Introduction to NoSQL

NoSQL

NoSQL (Not Only SQL)

- = NoSchema + NoTransactions + NoLanguage + NoStandards

= valaki, valahol nagyon utálja az SQL-t

Célkitűzés

- Elosztottság kezelése
  - Különböző objektumok különböző szerveren tárolódnak
  - Azonos objektumok replikációját meg kell oldani
- A hozzáférés késleltetését csökkentsük
  - Akár az ACID szabályok feláldozásával
- Fokozatos konzisztencia biztosítása
  - Előbb-utóbb mindenhova érjen el a változás
- Csak két (HTTP) művelettel legyen megvalósítható
  - PUT (Write)
  - GET (Read)

NoSQL nem egy konkrét termék

A NoSQL hat fő jellemzője

- Horizontális skálázhatóság (scale-out)
- Sokgépes környezet
- Egyszerű hívási felületek
- Konzisztencia részleges feladása
- RAM és elosztott indexek hatékony felhasználása
- Rugalmas sémaszerkezet

Típusai

- Kulcs-érték párok (pl. Voldemort, Dynamo, Dynomite)
- Oszlopcsaládok (BigTable leszármazottak: Hbase, HyperTable, Cassandra, PNUT)
- Gráf adatbázisok (pl. Neo4J, InfiniteGraph, AllegroGraph, InfoGrid)
- Dokumentumtárak (pl. CouchDB, MongoDB, SimpleDB)

RDBMS vagy NoSQL kell nekünk?

- RDBMS:
  - Struktúrált, szervezett adatok
  - SQL nyelv
  - Az adatok és kapcsolataik nem egy táblába tárolódnak
  - DCL, DDL, DML, Query (SELECT)
  - Szigorú adatkonzisztencia
- NoSQL:
  - Legtöbben Not Only SQL-ként értik
  - Nincs deklaratív szabvány-nyelv
  - Nincs előre definiált séma, struktúrálatlan adatok
  - Idővel beáll talán a konzisztencia, nem ACID
  - CAP és nem ACID
  - A prioritás a sebesség, rendelkezésre állás, skálázódás
  - BASE típusú tranzakciók

Elfogadjuk a CAP-tételt, de akkor legalább BASE

- A BASE típusú rendszer feladja a konzisztenciát, marad tehát a rendelkezésre állás és a performancia: lásd Facebook és hasonlók ☺
- BASE:
  - Basically Available
  - Soft state: az adat módosulhat input nélkül is
  - Eventual consistency: idővel helyreáll a konzisztencia ☺

Kritikák, megállapítások

A NoSQL megítélése ma „vallásvita”

- A NoSQL rávilágít arra, hogy az elmondottakat/tanultakat
  a) Ésszerűen alkalmazzuk
  b) A problémákra adekvát módon reagáljunk („one size fits all” nem igaz!).

  de a NoSQL azért nem hozott (eddig) **lényegesen** új elemeket
- Mi hozza/viszi a pénzt?
  - A felhasználók száma, türelme? (Facebook, IWIW)
  - A megbízhatóság, kiszámíthatóság? (Bank, Számlázás
- Skálázhatóság
  - NoSQL skálázható?
  - SQL nem skálázható (olcsón)?
- Divat
  - (mindenki forradalmár) …hetente indul új NoSQL projekt
  - (amúgy konzervatív) …de a fő vállalati rendszerek nem térnek át

Kritikák, megállapítások

A NoSQL megítélése ma „vallásvita”
(http://dx.doi.org/10.1145/1953122.1953144)

- „az SQL idejét múlt, alig fejlődik”
  - …de szinte mindenki meg tudja tanulni
  - alacsony szintű nyelvek (Erlang) idővel kihalnak
- ACID alapú RDBMS-ek teljesítménye „gyenge”
  - 23% naplózás
  - 20% zárkezelés
  - 11% kontextusváltás
  - 33% tárkezelés
- MapReduce
  - Nagyon hatékony…
  - …de még a Google is leváltotta

Továbbtanulási javaslat NoSQL irányban

- Nem vizsgatéma egyébként, de akinek többlet energiái vannak, annak javaslom, hogy nézzen utána a MongoDB-nek. Mostanában nagyon felfutóban van a „csillaga”
- A következő forrásokat ajánlom erre:
  - w3resource: Introduction to MongoDB
  - Tutorialspoint: MongoDB Tutorial
  - Megszámlálhatatlan egyéb tanulási lehetőség

- Azért ezek úgy működnek egy kicsit, mint a „divat”. Bármely pillanatban kikerülhet valaki a pikszisből és megjelenik az újabb „sztár” ☺