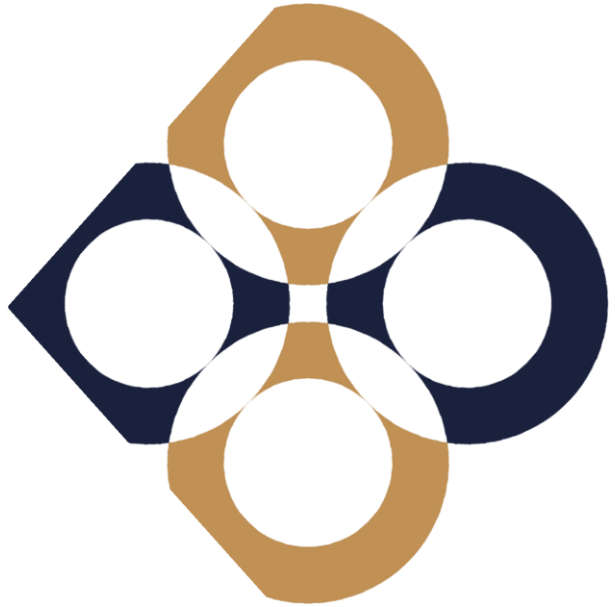
BUDAPESTI CORVINUS EGYETEM

Adatbázisok előadás 07

Nem relációs adatbázisok

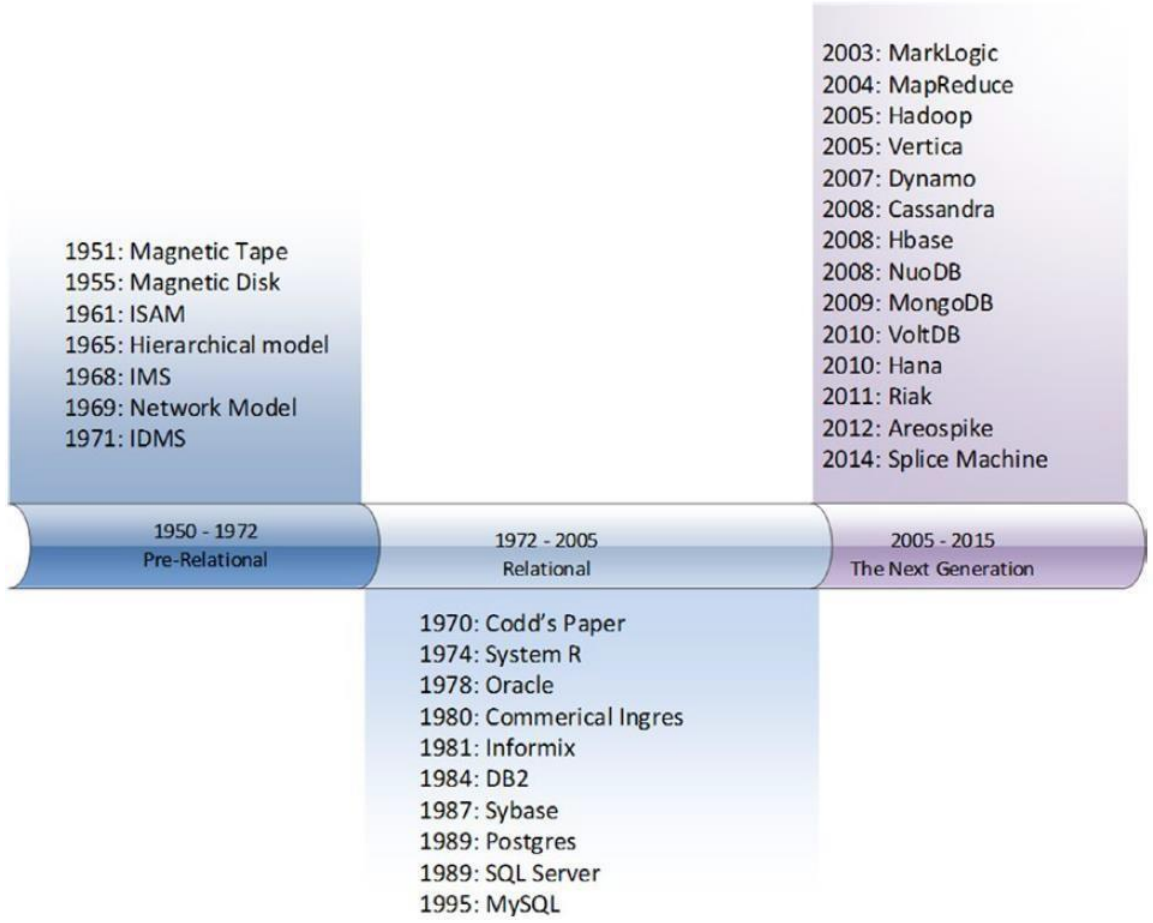


BUDAPESTI CORVINUS EGYETEM

Miről lesz szó?

- Relációs adatbázisok - kihívások
- Újgenerációs adatbázisok kialakulása
- NoSQL adatbázisok jellemzői
- In-memory adatbázisok
- Nevezetes relációs és nem relációs adatbázisok
- Választás az adatbázis rendszerek között
- Tipikus NoSQL adatbázisok elérése

Az adatbázisok fejlődése az időben



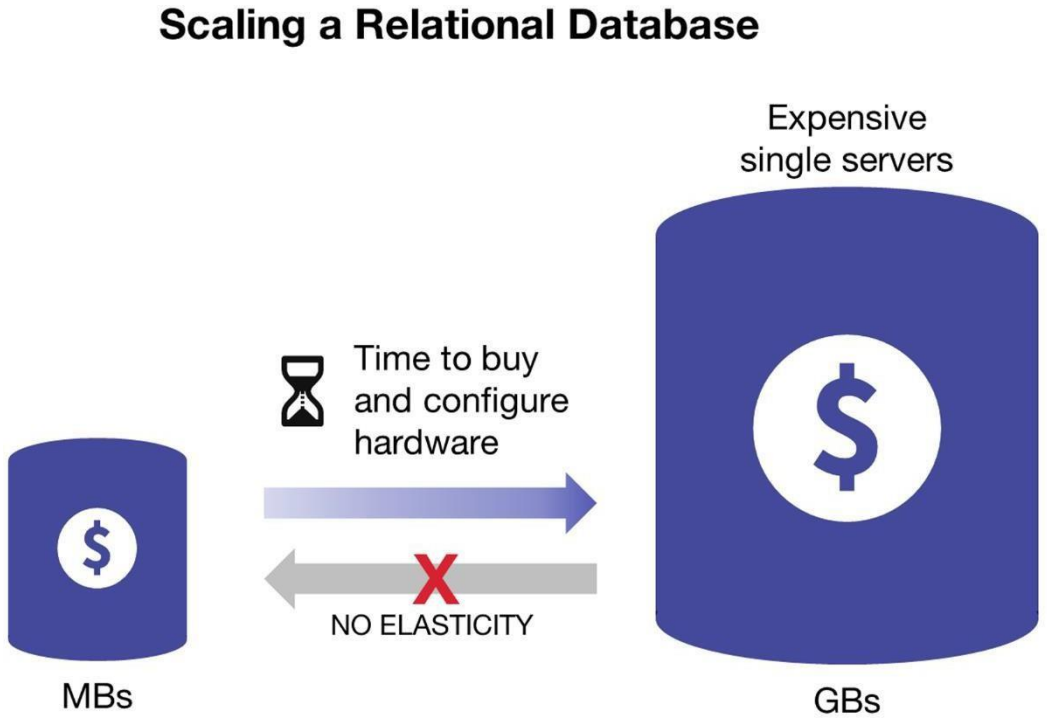
Három fő korszak*

- Korai adatbázisok
  - Hálós és hierarchikus adatmodell
  - Rugalmatlan struktúra
  - Nehézkes lekérdezés
- Relációs adatbázisok
  - Kliens-szerver architektúra
  - Relációs adatmodell
  - ACID tranzakciók
  - SQL nyelv
- **Újgenerációs adatbázisok**

* A fejlődés nem jelenti a korábbi adatbázisok eltűnését

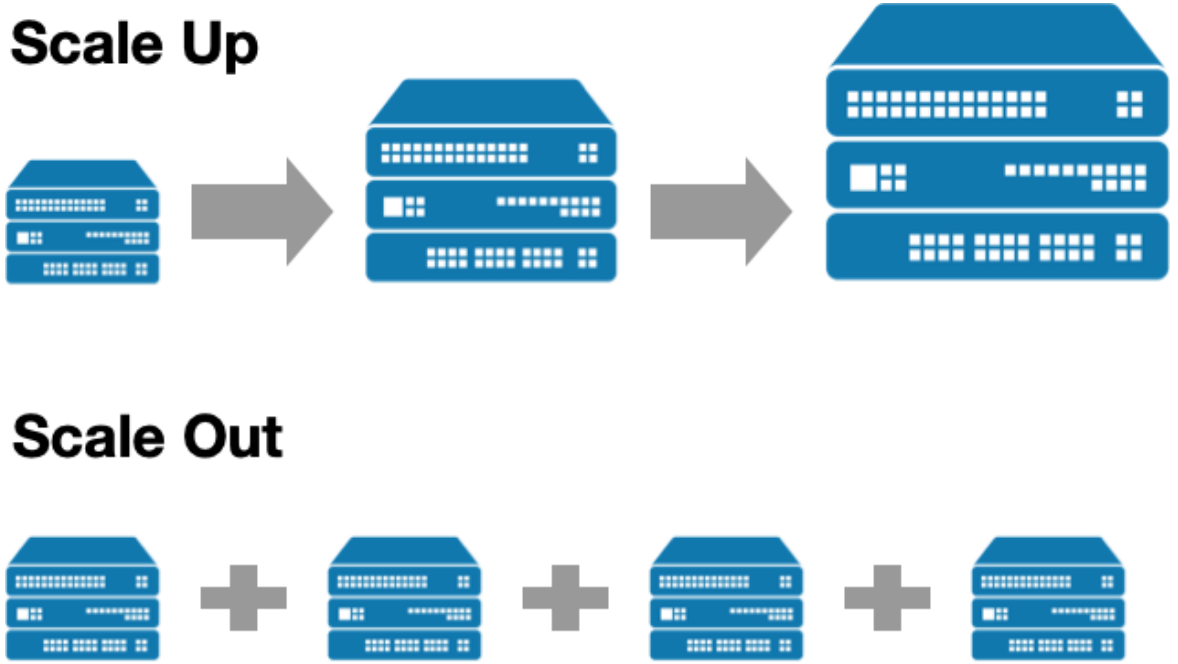
Relációs adatbázisrendszerek – Kihívások

- **Drága a skálázhatóság**
- Nagy mennyiségű adat real-time feldolgozása
- DBA kell a fenntartáshoz és a finomhangoláshoz
- Relációs adatokhoz tervezték

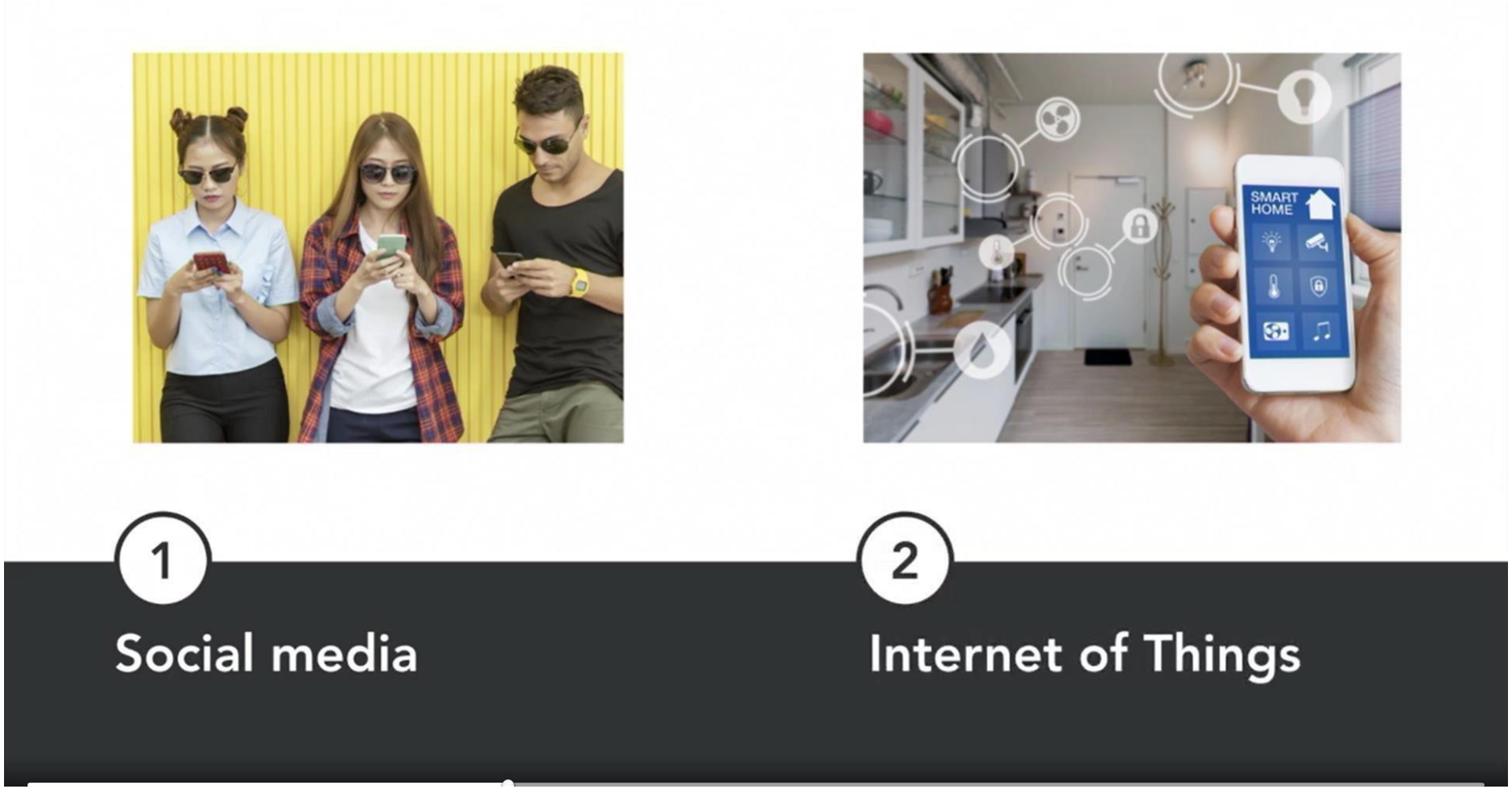


Skálázhatóság javítása – Scale Out

- **Scale Up** (függőleges skálázás)
  - valamely erőforrást (tároló, memória, processzor stb.) nagyobbra cserélünk
  - relációs adatbázisokra jellemző
- **Scale Out** (vízszintes skálázás)
  - több erőforrást adunk a rendszerhez, ezek egymással párhuzamosan működnek
  - Modern adatbázisokra jellemző

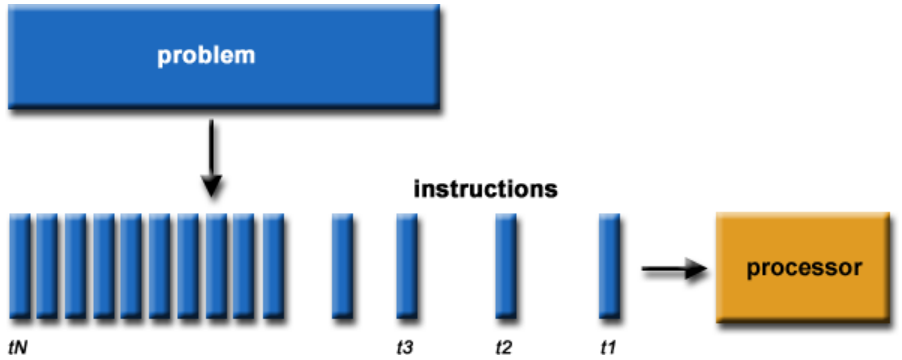


Az adatmennyiség növekedése

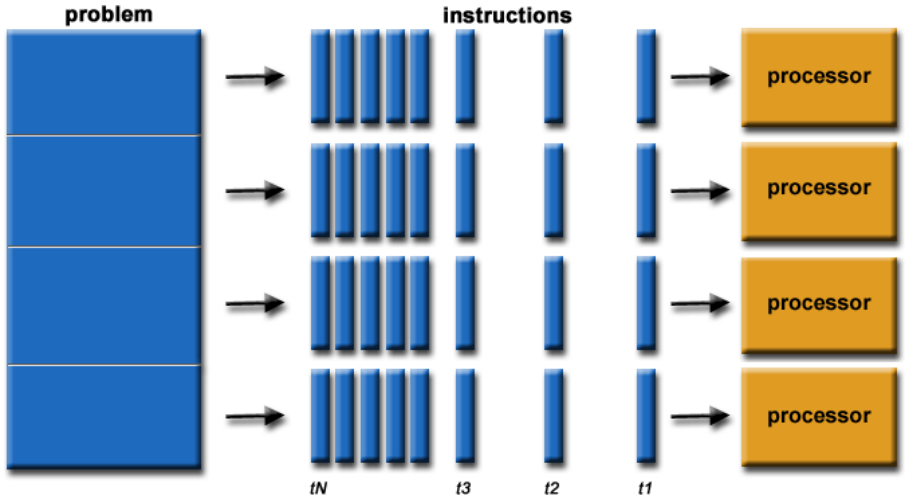


Nagy adatmennyiség feldolgozása – Párhuzamos számítások

- A párhuzamos számítás(parallel computing) a számítások olyan formája, amikor több számítás egyszerre hajtódik végre.
- Elve: a komplex problémák gyakran részekre bonthatók és a részek konkurens módon (párhuzamosan) oldhatók meg.



Soros végrehajtás



Párhuzamos végrehajtás

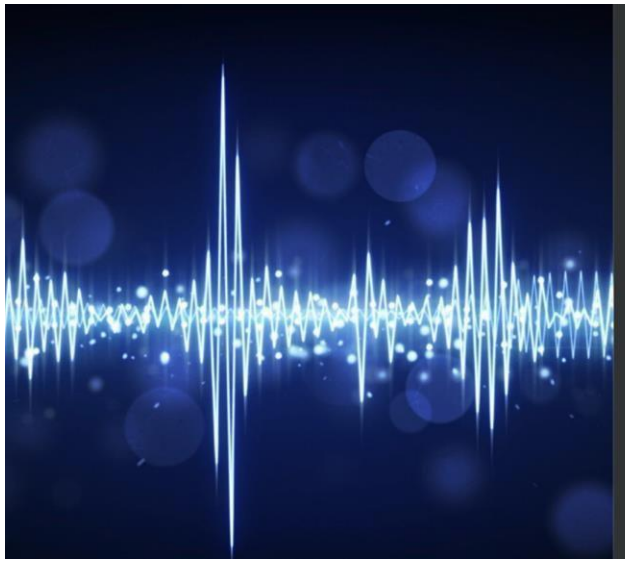
BUDAPESTI CORVINUS EGYETEM

Egyre több nem strukturált adat

Semi-structured

- Variables are marked with tags
- Tags can vary freely
- HTML
- XML
- JSON

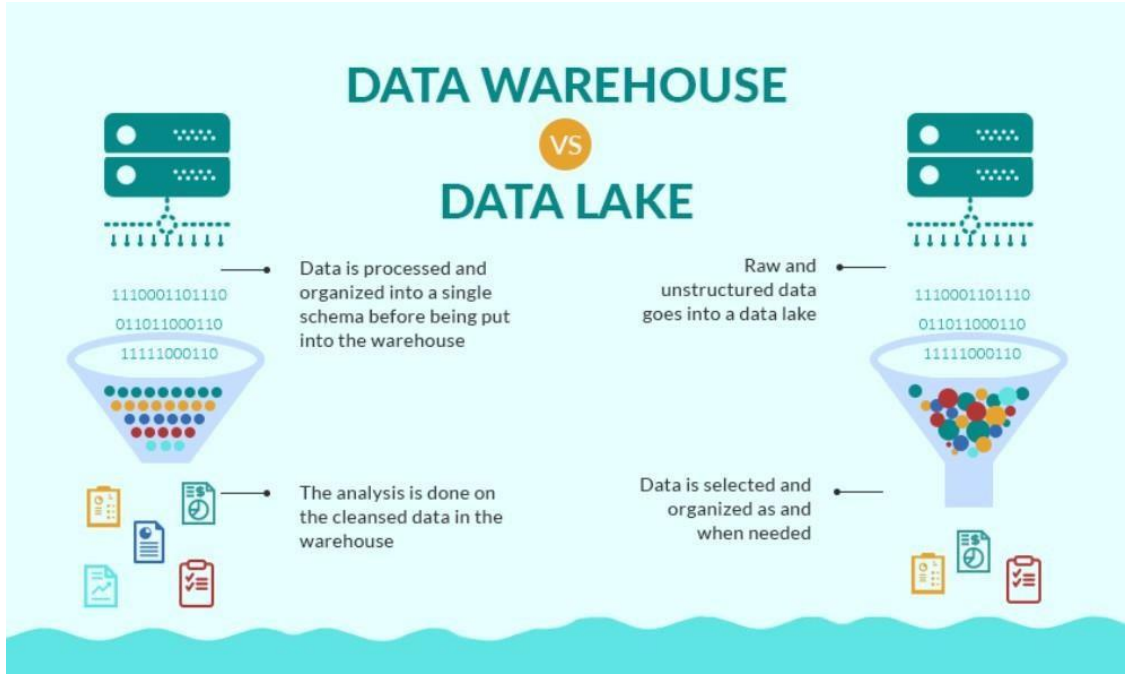

```
{
  "State": "California",
  "state_code": "CA",
  "region": "West",
  "governor": "Democrat",
  "psychRegions": "Relaxed and Creative",
  "extraversion": 51.4,
  "agreeableness": 49,
  "conscientiousness": 43.2
},
{
  "State": "Colorado",
  "state_code": "CO",
  "region": "West",
  "governor": "Democrat",
  "psychRegions": "Friendly and Conventional",
  "extraversion": 45.3,
  "agreeableness": 47.5,
  "conscientiousness": 58.8
},
```

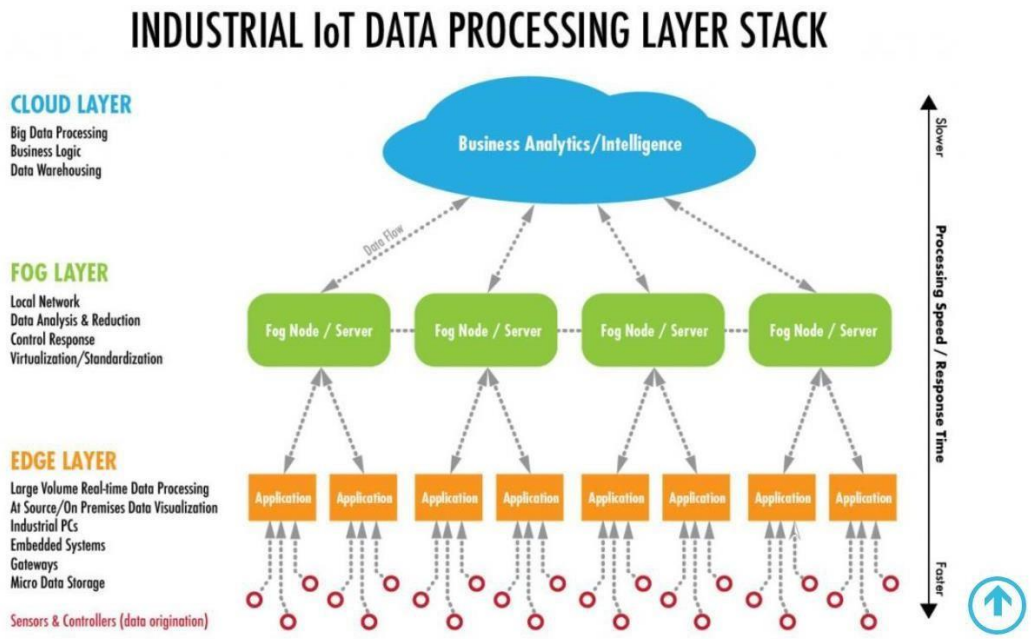



Unstructured

- Variables and fields are not labeled or identified
- Free text
- Photos
- Video
- Audio

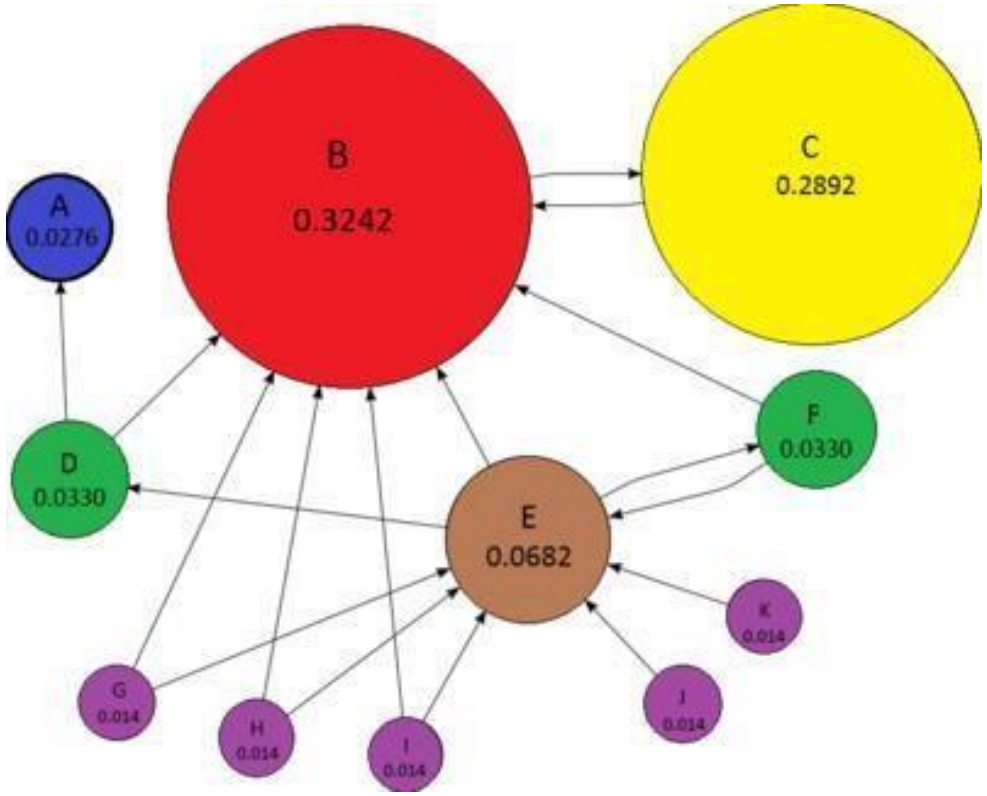
Adattárolás/feldolgozás - Data lake, Cloud, Fog, Edge





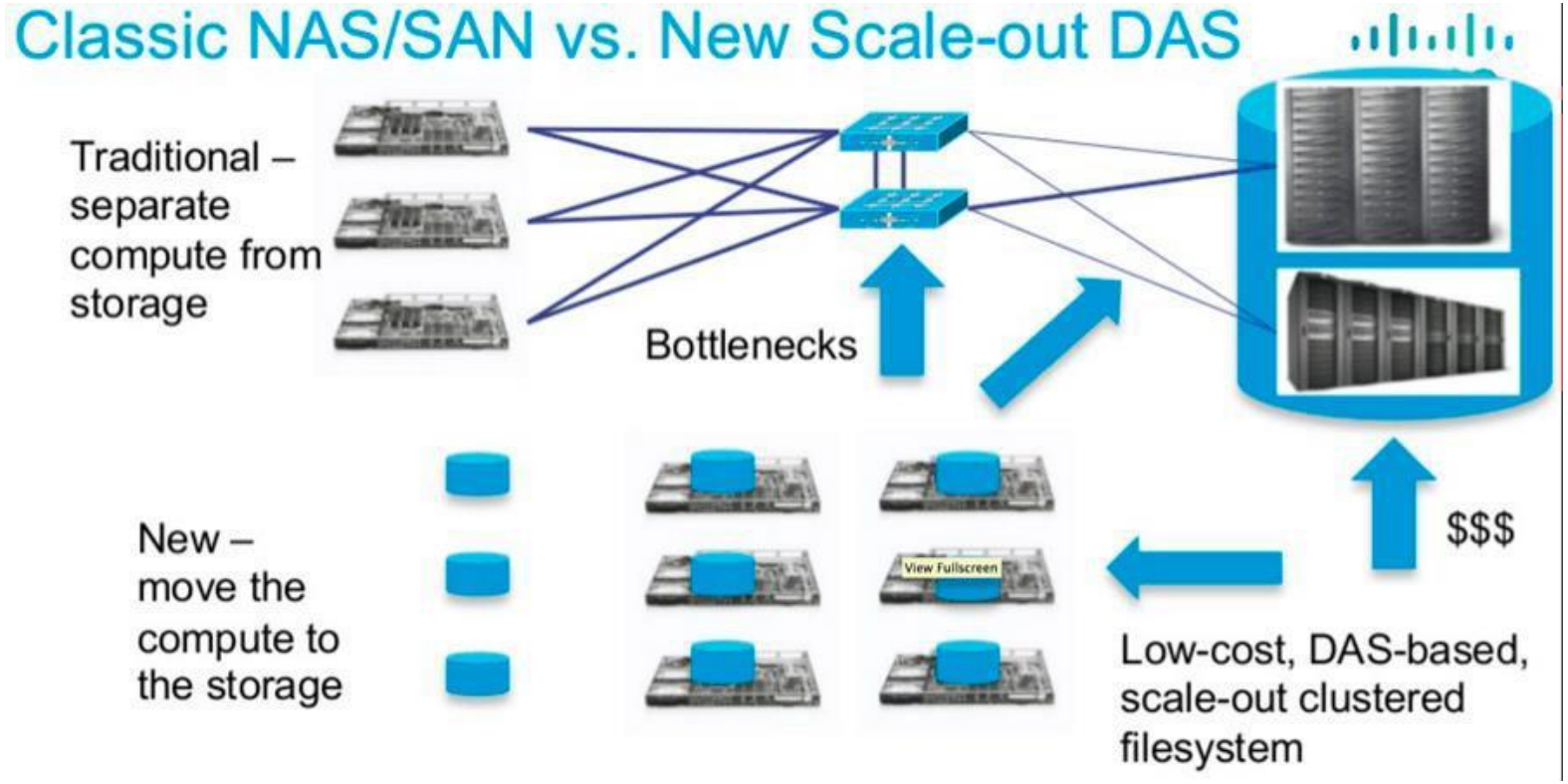
Újgenerációs adatbázisok – A Google szerepe - PageRank

- Keresőmotor – PageRank algoritmus (1997)
  - Egy weboldal fontossága, besorolása (ranking) attól függ, hogy hány másik oldal hivatkozik rá, illetve a hivatkozó oldalak milyen besorolásúak
  - Intelligensen alkalmazkodik a növekvő adatmennyiséghez (machine learning)
  - Sokkal eredményesebb a hagyományos (indexeket használó) keresőmotoroknál



https://de.wikipedia.org/wiki/PageRank

Újgenerációs adatbázisok – A Google szerepe - Hardver

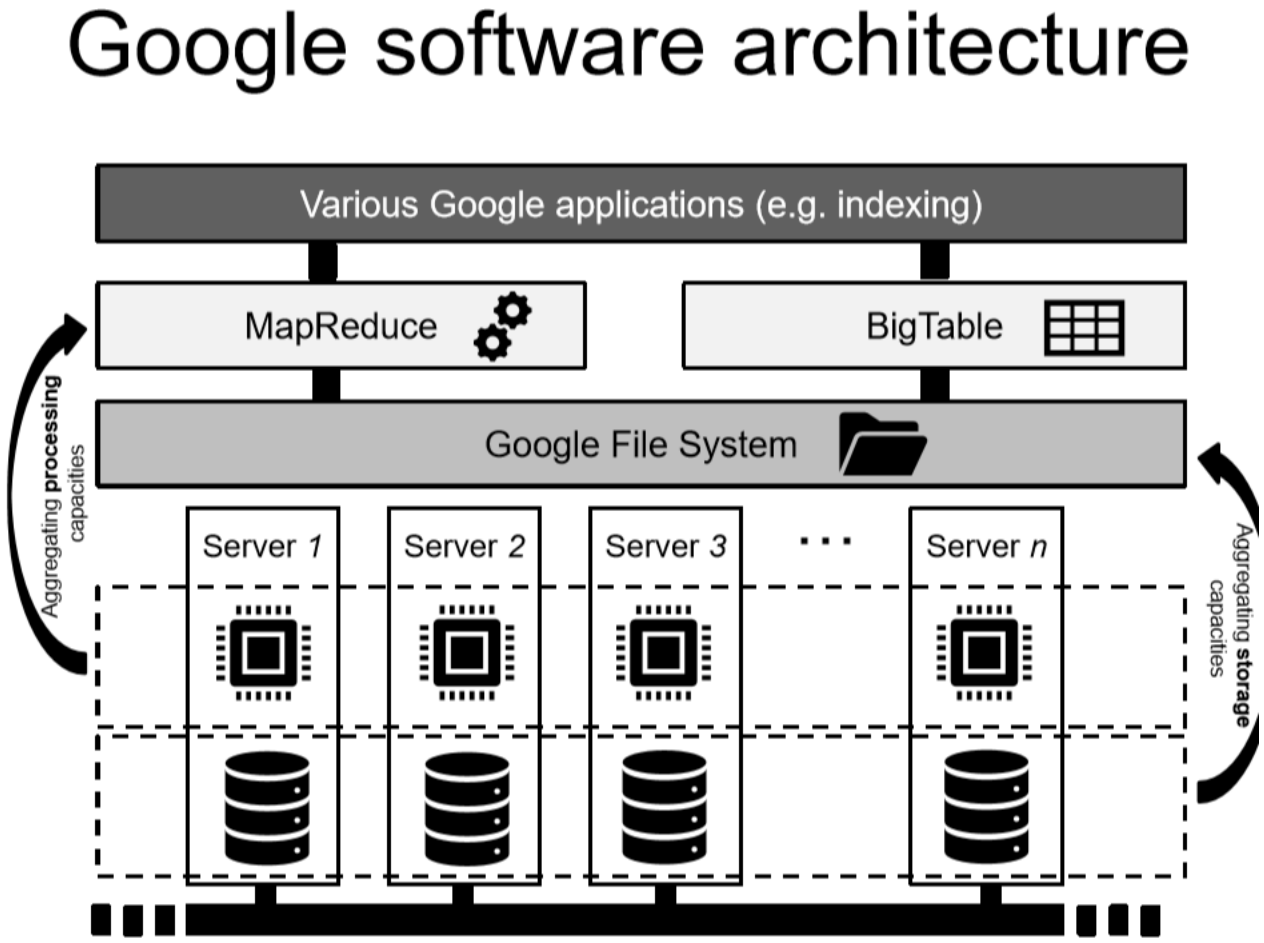


- Nagy számú, alacsony költségű szerver
- Moduláris felépítésű szerverek
- Közvetlenül csatlakoztatható tároló lemezek (DAS, Direct attached storage)
- A számítási kapacitás dinamikusan növelhető

https://0x0fff.com/hadoop-on-remote-storage/

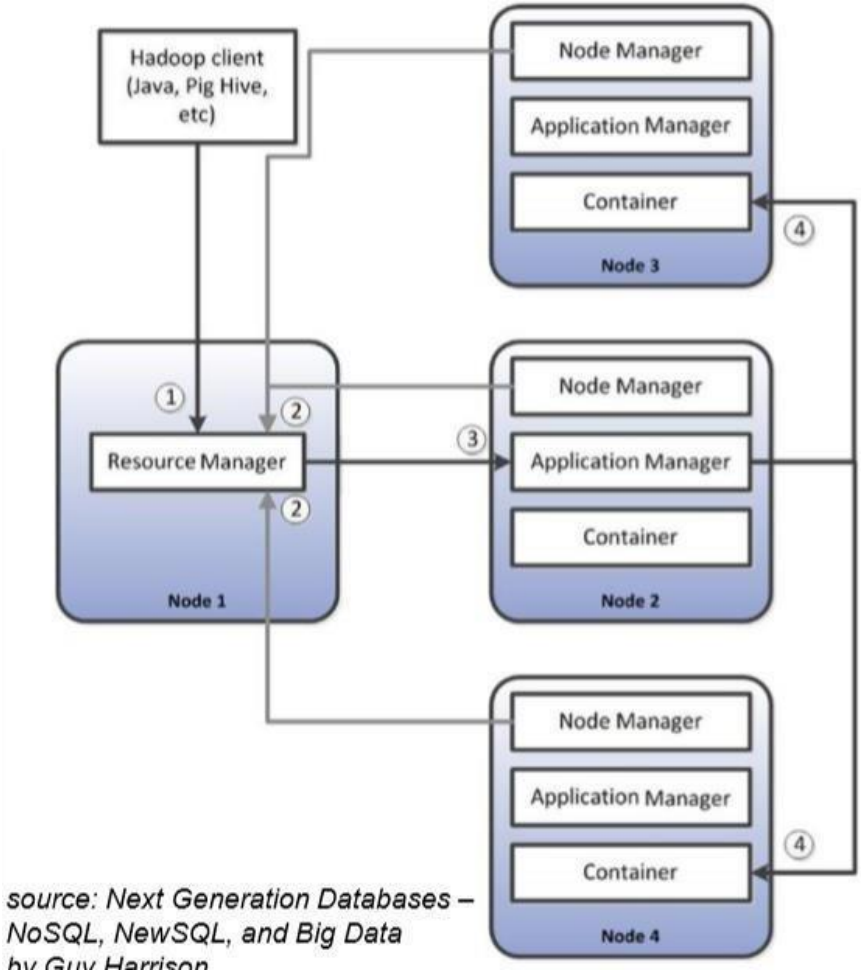
Újgenerációs adatbázisok – A Google szerepe - Szoftver

- **Google File System (GFS)**: elosztott fájlrendszer
- **MapReduce**: elosztott keretrendszer párhuzamos számításokra
- **BigTable**: nemrelációs adatbázis



Újgenerációs adatbázisok – Hadoop

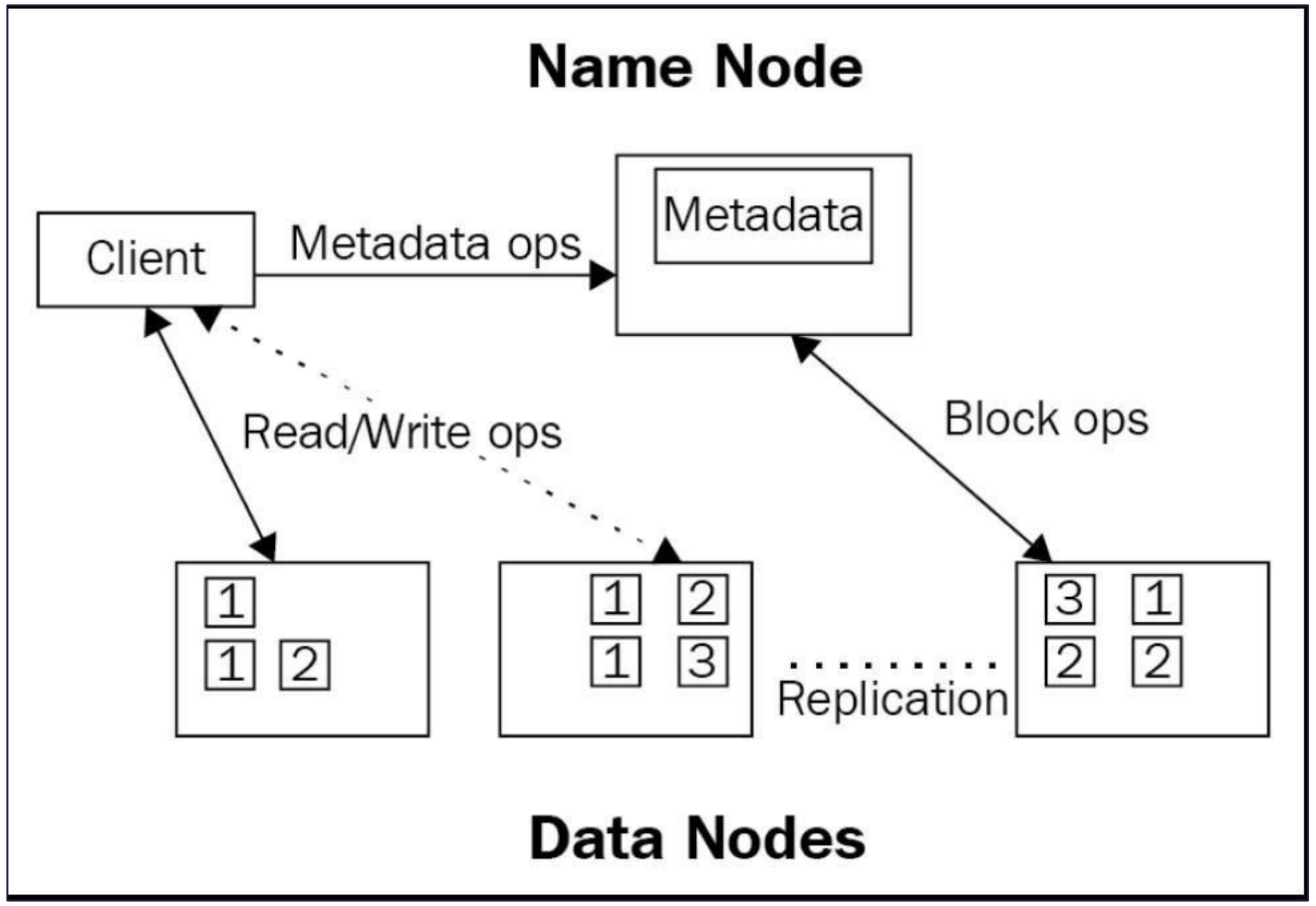
- A Google architektúra open-source megvalósítása
- Robosztus és skálázható Big data platform
- HDFS (Hadoop File System) fájlrendszer
- Adatok betöltése séma definíció nélkül (schema on read)



source: Next Generation Databases – NoSQL, NewSQL, and Big Data by Guy Harrison

Hadoop Distributed File System

- A fájlokat adott méretű blokkokra osztja (128 MB)
- A blokkok replikálva vannak (általában 3-szor)
- Két fő komponens
  - Name Node
  - Data Node
- Hibatűrő és hiba helyreállító
- Főleg kötegelt feldolgozás
- Nem cél az adatok gyors elérése



Hadoop és a relációs adatbázisok összehasonlítása

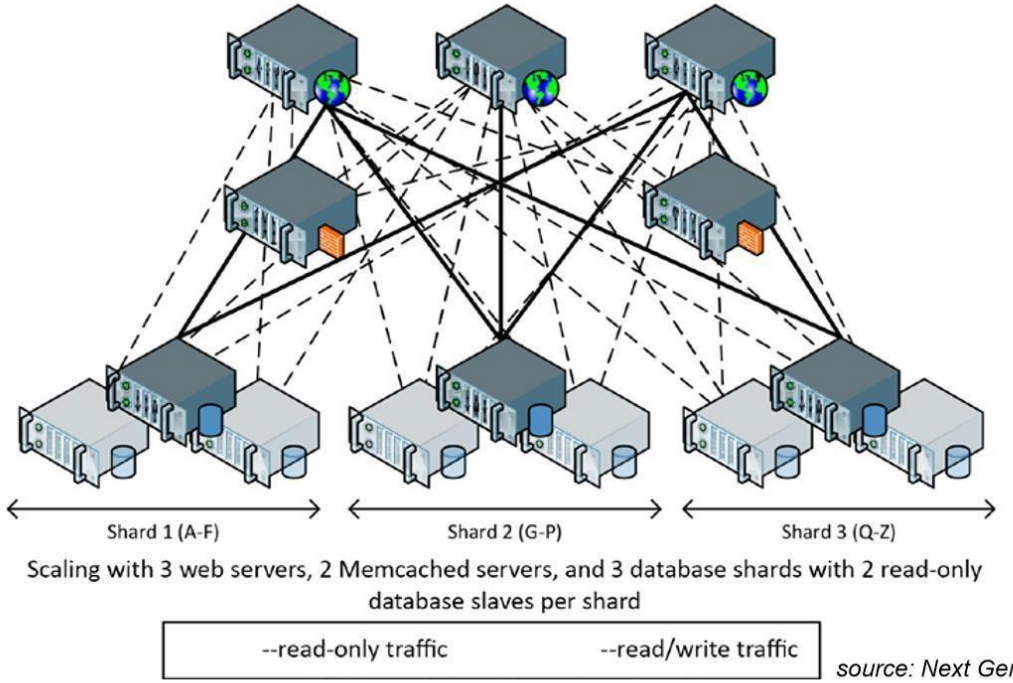
Example Comparison: RDBMS vs. Hadoop

	Traditional RDBMS	Hadoop / MapReduce
Data Size	Gigabytes (Terabytes)	Petabytes and greater
Access	Interactive and Batch	Batch – NOT Interactive
Updates	Read / Write Many Times	Write Once, Read Many Times
Structure	Static Schema	Dynamic Schema
Integrity	High (ACID)	Low
Scaling	Nonlinear	Linear
Query Response Time	Can Be Near Immediate	Has Latency (Due to Batch Processing)

Play

Újgenerációs adatbázisok – A közösségi oldalak szerepe- Sharding

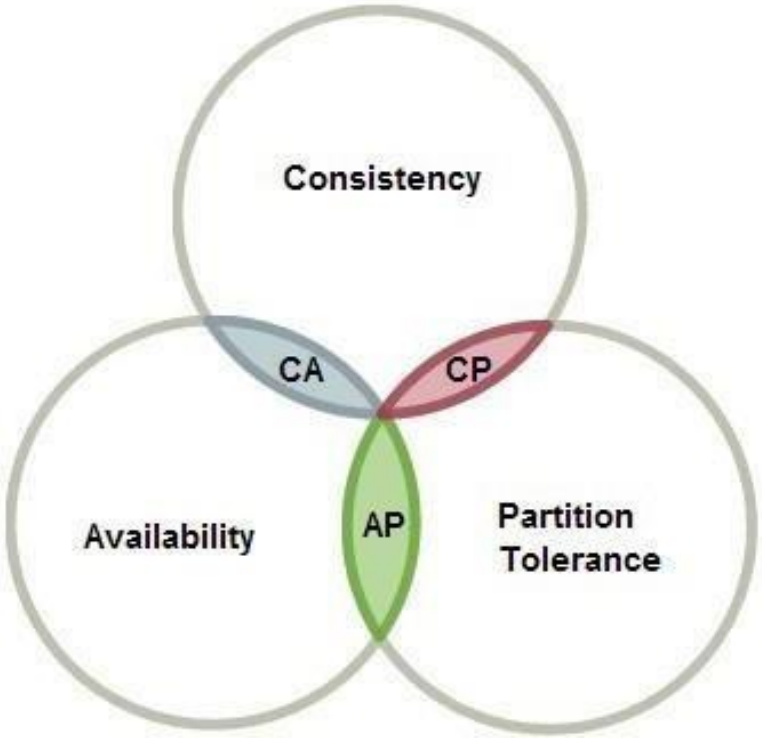
- A nagyobb táblák részekre vannak bontva (partíciók)
- A partíciók több szerveren tárolódnak
- A partíciókat shard-eknek (szilánk) is nevezzük
- A partícionálás kulcsértékek alapján történik
- Az olvasást egy elosztott gyorsítótár (Memcached) segíti
- Többnyire MySQL szerverek valósítják meg



source: Next Generation Databases – NoSQL, NewSQL, and Big Data by Guy Harrison

Sharding – a tranzakciós integritás elvesztése

- Az ACID tranzakciókezelés itt nem valósítható meg
- Helyette CAP
  - Consistency
  - Availability
  - Partition tolerance
- Az elosztott adatbázis rendszerekben a CAP kritériumok közül egyszerre max. 2 teljesülhet (**CAP Theorem**)



BUDAPESTI CORVINUS EGYETEM

SQL és NoSQL adatbázisok összehasonlítása

Szempont	SQL	NoSQL
Adatbázis rendszer	Relációs	Elosztott
Adatok	Strukturált	Strukturált, nem strukturált és félig strukturált
Komplex lekérdezések	Alkalmas	Korlátozottan alkalmas
Rugalmasság	Rugalmatlan	Rugalmas
Séma	Előre definiált	Dinamikus
Skálázhatóság	Vertikális	Horizontális
Tranzakciók	ACID	CAP

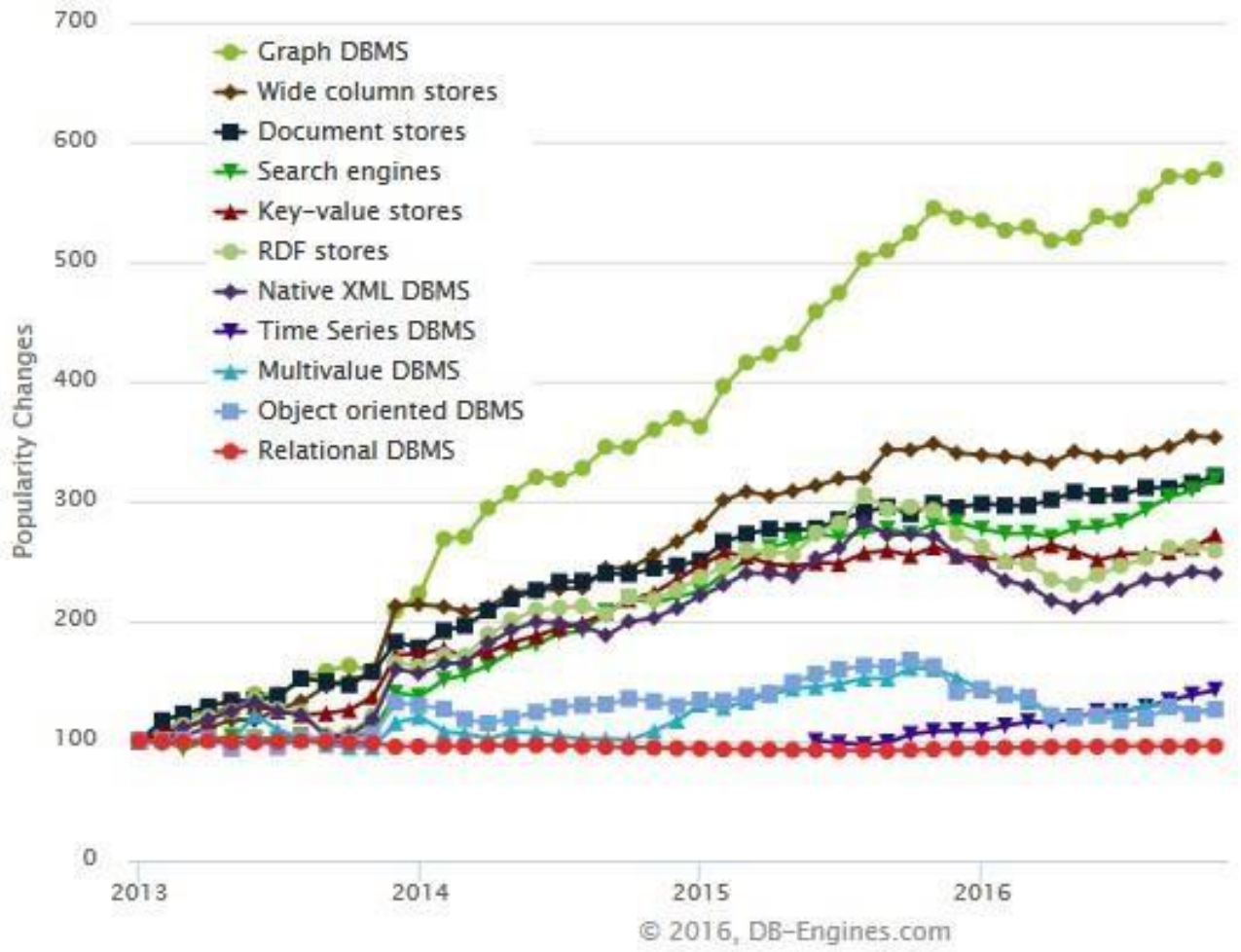
NoSQL (Not only SQL) adatbázisok

- Az elnevezés az újgenerációs adatbázisokra utal, ahol a lekérdezés többnyire nem SQL nyelven történik
- Főbb típusai
  - Dokumentum adatbázisok
  - Gráf adatbázisok
  - Oszlopalapú adatbázisok
  - Kulcs-érték adatbázisok

https://www.getfilecloud.com/blog/2014/08/leading-nosql-databases-to-consider/

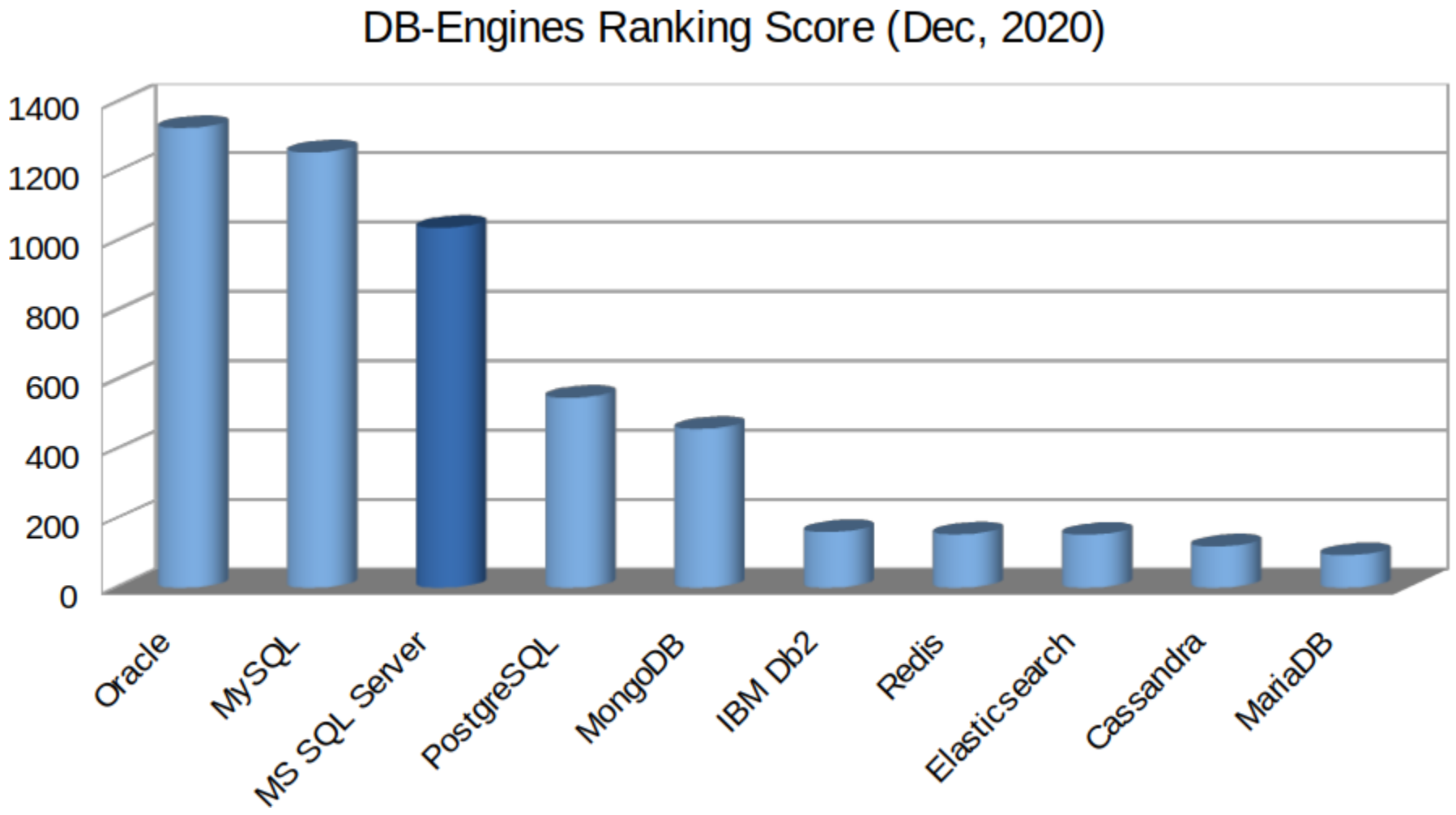
BUDAPESTI CORVINUS EGYETEM

Trendek – adatbázis modellek



BUDAPESTI CORVINUS EGYETEM

Piacvezető adatbáziskezelő rendszerek



Oracle

Az Oracle Corporation terméke, alkalmas relációs és NoSQL adatbázisok, elosztott adatbázisok létrehozására és kezelésére

- Előnyök
  - Sokféle igénynek megfelel – pl. adattárház, big data, OLTP
  - Felhős és on-premise változat is létezik
  - Nagy teljesítmény, skálázhatóság, magas rendelkezésre állás, biztonság
  - Terméktámogatás, felhasználói közösségek
- Hátrányok
  - KKV-kszámára nem ideális a magas költségek miatt
  - Nem egyszerű bevezetni
  - Erőforrás igényes

MySQL

Open-source adatbázis rendszer, amely sokféle platformra telepíthető.

- Előnyök
  - Nagyméretű táblák támogatása
  - Könnyen megtanulható, testreszabható
  - Gyors, skálázható, biztonságos
  - Felhasználói közösségek
- Hátrányok
  - Atranzakciók kezelése nem elég hatékony
  - Atulajdonos az Oracle – a fejlesztést így nem a közösség irányítja
  - Stabilitási kérdések
  - Nehézkes skálázhatóság

BUDAPESTI CORVINUS EGYETEM

Microsoft SQL Server

Microsoft SQL Server

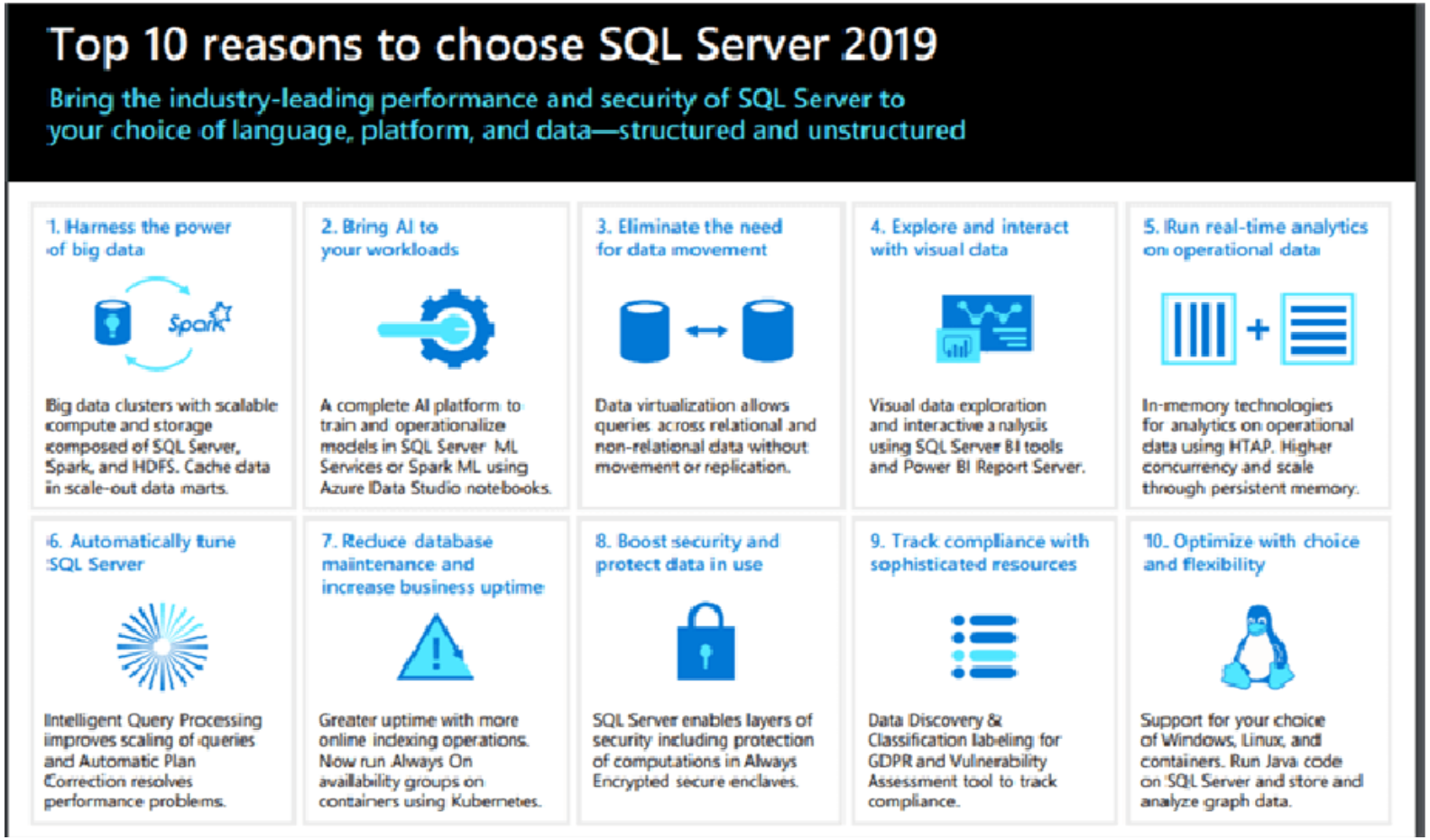
A legismertebb kereskedelmi célú adatbázis rendszer Windows platformon

- Előnyök
  - Sokféle igénynek megfelel
  - Kiemelkedően jó terméktámogatás
  - Sokféle kiegészítő eszköz és alkalmazás
  - Felhős és on-premise változat
  - Integráció a többi MStermékkel
- Hátrányok
  - Nem olcsó
  - Bonyolult licenszelés
  - Alekérdezések optimalizálása nem egyszerű
  - Aforráskód verziókezelése nem támogatott

BUDAPESTI CORVINUS EGYETEM

Microsoft SQL Server

Microsoft SQL Server



BUDAPESTI CORVINUS EGYETEM

PostgreSQL

Népszerű nyílt forráskódú adatbáziskezelő rendszer, használja többek között az Apple, Cisco, Fijutsu, Skype

- Előnyök
  - Sokféle platformon működik
  - Ingyenes
  - Külső adatforrások könnyű integrálása
  - Nagyméretű táblák, táblaterek, tranzakció kezelés
- Hátrányok
  - Nem egyszerű a konfigurálása
  - Replikáció kezelése

BUDAPESTI CORVINUS EGYETEM

MongoDB

Alegnépszerűbb NoSQLadatbázis rendszer, amely többféle programozási nyelvet is támogat.

- Előnyök
  - Nagy teljesítményű, gyors, séma nélküli adatbázis
  - Indexelhető mezők
  - Auto-sharding
  - Könnyű adminisztráció
- Hátrányok
  - Nincsenek JOIN-ok
  - Az adatok mérete nagy
  - Memória használat

BUDAPESTI CORVINUS EGYETEM

IBM DB2

IBM DB2

Népszerű relációs adatbázis rendszer, az Oracle egyik fő vetélytársa

- Előnyök
  - Többféle platformon fut
  - Integráció más IBM termékekkel
  - Adatmigráció támogatás MS SQL és Oracle rendszerekből
  - NoSQL adatmodellt istámogat
  - Felhős és on-premise változat is elérhető
- Hátrányok
  - Magas költségek
  - Korlátozott támogatás

BUDAPESTI CORVINUS EGYETEM

Microsoft Access

Irodai, kisvállalati környezetben elterjedt relációs adatbázis

- Előnyök
  - Egyszerű telepítés
  - Integrált grafikus környezet
  - VBA-támogatás
  - Olcsó
  - Import/export sokféle formátummal
- Hátrányok
  - Korlátozott képességek
  - Mindent egy fájlban tárol
  - Bugok magas száma

BUDAPESTI CORVINUS EGYETEM

Cassandra

Népszerű oszlop-alapú NoSQL adatbázis, többek között a Facebook és a Twitter is használja

- Előnyök
  - Gyors válaszidők
  - ACID tulajdonságok
  - Flexibilis, skálázható, egyenrangú (peer to peer) architektúra
  - Integráció más Java-alapú alkalmazásokkal
- Hátrányok
  - A teljesítmény előre nem látható
  - Ad-hoc lekérdezéseket nem támogatja
  - Az aggregációkat korlátozottan támogatja

BUDAPESTI CORVINUS EGYETEM

Redis

Kulcs-érték tároló NoSQLadatbázis, amely sokféle programnyelvet támogat

- Előnyök
  - Könnyű telepítés
  - Sokféle adattípus támogatása
  - Gyors
  - Hibatűrés
  - Az egész adatbázis a memóriában van
- Hátrányok
  - JOIN-okat nem támogatja
  - Az adatoknak el kell férni amemóriában

BUDAPESTI CORVINUS EGYETEM

Elasticsearch

elastic

Open-source dokumentum-orientált kereső motor (adatbázis), amelyet használ pl. a Wikipedia, The Guardian, StackOverflow, GitHub

- Előnyök
  - Fejlett szövegkeresési képességek
  - Sémafüggetlen
  - Skálázható - akár petabyte méretű strukturált és nem strukturált adatok
  - Saját API
  - Nagymértékben elosztható
- Hátrányok
  - Nem támogat több nyelvet

BUDAPESTI CORVINUS EGYETEM

Hogyan válasszunk adatbáziskezelő rendszert?

Főbb szempontok, amiket mérlegelni kell

- Technikai szempontok
  - Adatmodell – relációs vs. NoSQLvs.In-Memory
  - Adatok konzisztenciája, adatbiztonság, adatvédelem
  - Egyidejű hozzáférések, integrálhatóság
  - Hatékonyság, használhatóság
- Költség szempontok
  - Szoftver ára, fenntartási költségek, hardver ára
  - Telepítési és működtetési költség, képzési költség
- Egyéb szempontok
  - Szervezeti kultúra, üzletpolitika

BUDAPESTI CORVINUS EGYETEM

Mikor használjunk relációs adatbáziskezelő rendszert?

Tipikus használati esetek

- Ha fontos az ACID tranzakció kezelés
- Ha a séma fix
- Ha (többnyire) strukturált adatokkal dolgozunk
- Ha az adatok között nincs social media, IoT, streaming audio stb. jellegű adat
- Ha a skálázhatóság nem kritikusan fontos
- Pl: OLTP rendszerek, batch processing, üzleti intelligencia

BUDAPESTI CORVINUS EGYETEM

Mikor használjunk NoSQL adatbáziskezelő rendszert?

Általános esetben akkor, ha decentralizált, skálázható, hibatűrő adatbázis rendszert szeretnénk kiépíteni, amely képes kezelni nem strukturált vagy félig strukturált adatokat is.

Aválasztásnál vegyük figyelembe a következőket is a NoSQLadatbázisokkal kapcsolatban

- ❑ AzSQLlekérdező nyelv (többnyire) nem használható
- ❑ Atranzakciók általában nem tesznek eleget azACID tulajdonságoknak
- ❑ Nincs egységes megközelítés, sok különböző modell
- ❑ Gyorsan változó körülmények

BUDAPESTI CORVINUS EGYETEM

Milyen NoSQL adatbázist válasszunk?

Tipikus használati esetek

- Kulcs-érték tárolók
  - Ha fontos a magas rendelkezésre állás, a flexibilis séma és a gyors válaszidő
  - Ha az adatok között nincsen sok, komplex kapcsolat
  - Pl: multimédia fájlok, felhasználói profilok, játékok, reklámok
- Dokumentum tárolók
  - Ha többféle szempont szerint szeretnénk lekérdezni
  - Ha sok valós idejű adatot kell kezelni
  - Ha fontos a flexibilis séma, és a gyors fejlesztés
  - Pl: logolás, online vásárlás, tartalom kezelés

BUDAPESTI CORVINUS EGYETEM

Milyen NoSQL adatbázist válasszunk?

Tipikus használati esetek

- Oszlop-alapú adatbázisok
  - Ha ritkán írunk, inkább olvasunk (lekérdezünk)
  - Ha lekérdezéskor többnyire oszlopokban gondolkodunk
  - Ha adatelemzési, adatbányászati feladatot kell megoldani
  - Pl: logolás, tartalom kezelés, adattárházak
- Gráf adatbázisok
  - Ha az adatok között sok kapcsolat van
  - Ha az adatok viszonylag ritkán változnak
  - Pl: közösségi média, útvonaltervek, online ajánlatok

BUDAPESTI CORVINUS EGYETEM

Mikor használjunk In-memory adatbáziskezelő rendszert?

Tipikus használati esetek

- Ha az adatbázis elfér a memóriában
- Ha fontos a teljesítmény és a gyors válaszidő
- Ha adatokat szeretnénk elemezni
- Ha valós idejű alkalmazást fejlesztünk
- Ha a tranzakciók száma magas
- Ha nem probléma, hogy a hardver költséges lehet
- Pl: pénzügyi alkalmazások, real-time piaci elemzések, kockázat elemzés

BUDAPESTI CORVINUS EGYETEM

Hogyan érhetők el a NoSQL adatbáziskezelők?

- Felhős megoldások*
  - Microsoft Azure
  - Google Cloud
  - Amazon Web Services
- Online Sandbox-ok
- On-premise megoldások

Dokumentum adatbázis – MongoDB

- Nyílt forráskódú adatbázis
- Dokumentum-orientált adatmodell
- Futtatási/telepítési lehetőségek
  - Mongo playground (https://mongoplayground.net/)
  - MongoDB Atlas (https://account.mongodb.com/account/login)
  - Microsoft Azure (Kezdőlap - Microsoft Azure)

    On-premise (Install MongoDB Community Edition — MongoDB Manual)

BUDAPESTI CORVINUS EGYETEM

Gráf adatbázis – Neo4J

- Nyílt forráskódú adatbázis
- Gráf adatmodell
- Futtatási/telepítési lehetőségek
  - Online sandbox (https://neo4j.com/sandbox/)

    On-premise (Neo4j Desktop Download - Launch and Manage Neo4j Databases)
    - Telepíthető Linux, Windows és Mac alá is
- Cloud – Neo4J Aura

Kulcs-érték adatbázis – Redis

- Nyílt forráskódú adatbázis
- Kulcs-érték adatmodell, in-memory
- Futtatási/telepítési lehetőségek
  - Cloud próba (Try Redis)
  - On-premise

    Redis - Linux

    How To Install Redis on Windows 10 - DEV Community – Windows

    Install and config Redis on Mac OS X via Homebrew | by Pete Houston | Medium - Mac

BUDAPESTI CORVINUS EGYETEM

Oszlopalapú adatbázis – Cassandra

- Nyílt forráskódú adatbázis
- Oszlopalapú adatmodell
- Futtatási/telepítési lehetőségek
  - Online proba (Try It Out | Datastax)
  - On-premise (Download (apache.org))
    - Telepíthető Linux, Windows és Mac alá is
- Cloud
  - Azure Cosmos DB
  - DataStax Astra

Források

- Guy Harrison: Next Generation Databases (Apress, 2015)

Köszönöm a figyelmet!