



Računarstvo i informatika

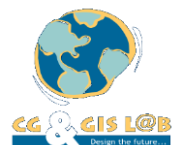
Katedra za računarstvo

Elektronski fakultet u Nišu

Sistemi baza podataka

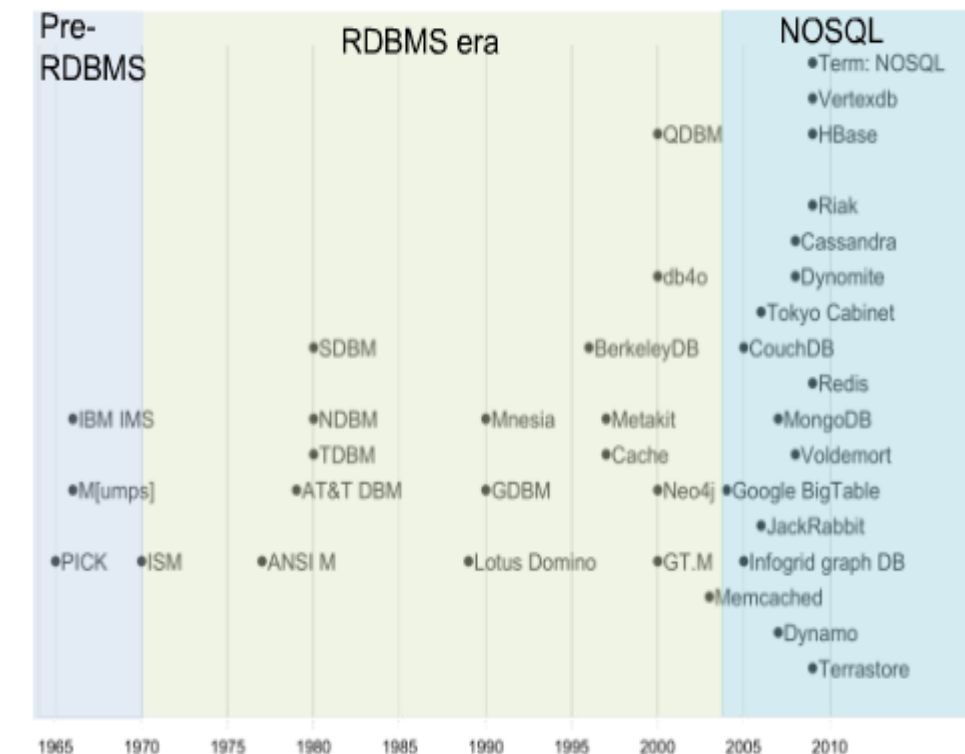
NoSQL baze podataka

Letnji semestar 2014/2015



Uvod

- Skladišta podataka pre nastanka RDBMS.
- Skladišta bazirana na hijerarhiji i multideimnzionalnim poljima.
- RDBMS – najpopularnije i generičko rešenje za skladištenje podataka.
- Sa ubrzanim razvojem Web aplikacija javila se potreba za drugačijim tehnikama za skladištenje podataka – NoSQL baze podataka.





Uvod

- Relacioni model je predstavljen 1970 godine
- E. F. Codd, “**A Relational Model of Data for Large Shared Data Banks**”
- Relaciona algebra obezbeđuje deklarativne mehanizme za rad sa skupovima podataka.
- SQL se bazira na relacionoj algebri.

Ime	Prezime	Indeks	MBR
Petar	Petrović	1111	123456
Milan	Milanović	2222	654321
Jovan	Jovanović	3333	345612



Uvod

- Prednosti korišćenja RDBMS:
 - Efikasno skladištenje podataka
 - Podrška za ACID transakcije
 - Podrška za kompleksne SQL upite
 - Ogromna tehnološka baza (različiti DBMS-ovi, alati, programski interfejsi i sl.)

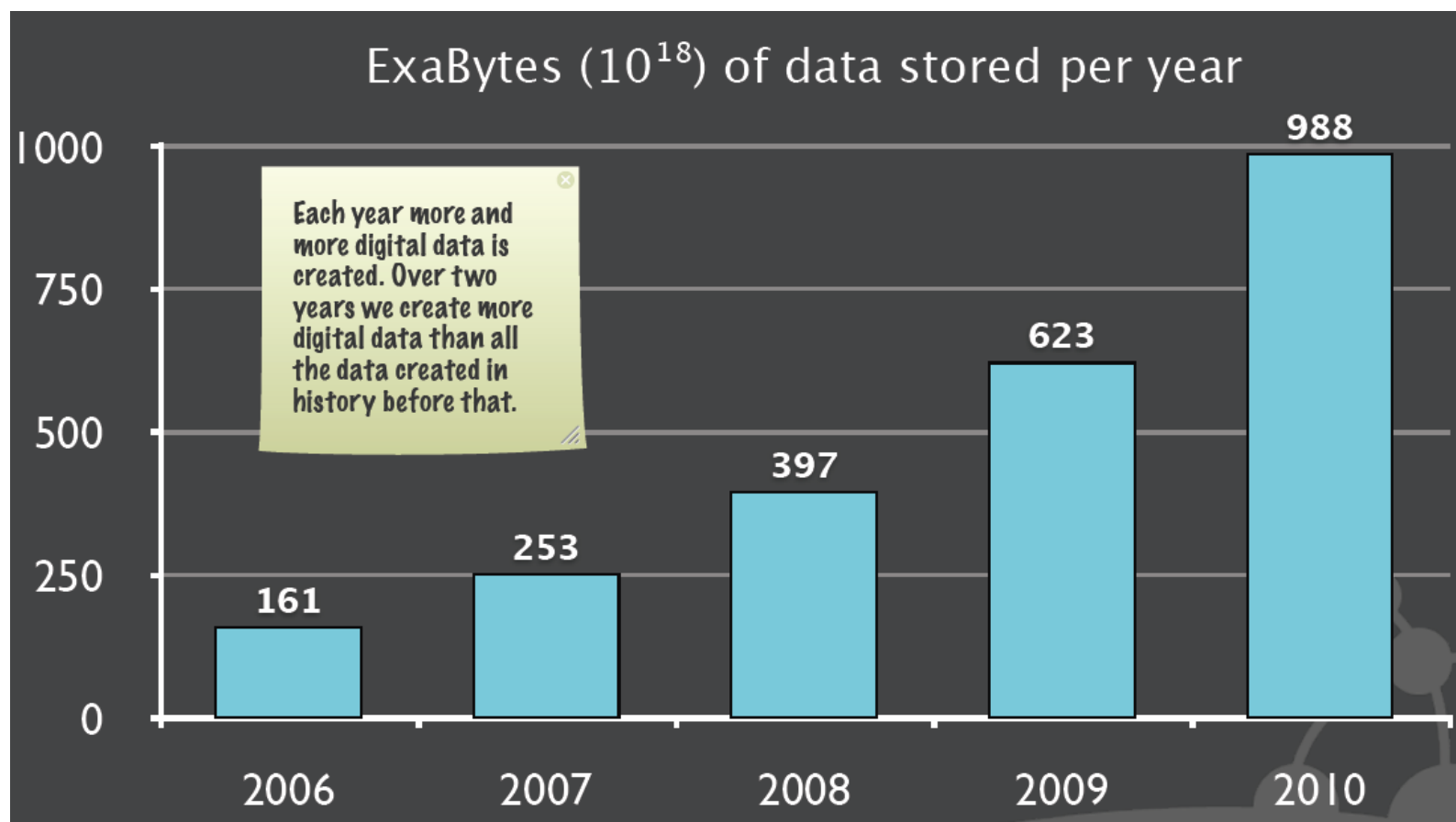


Podaci na Web-u

- Četiri osnovne karakteristike podataka na Web-u:
 - Količina podataka
 - Povezanost podataka (relacije)
 - Polustrukturiranost podataka
 - Arhitektura aplikacija koje koriste podatke

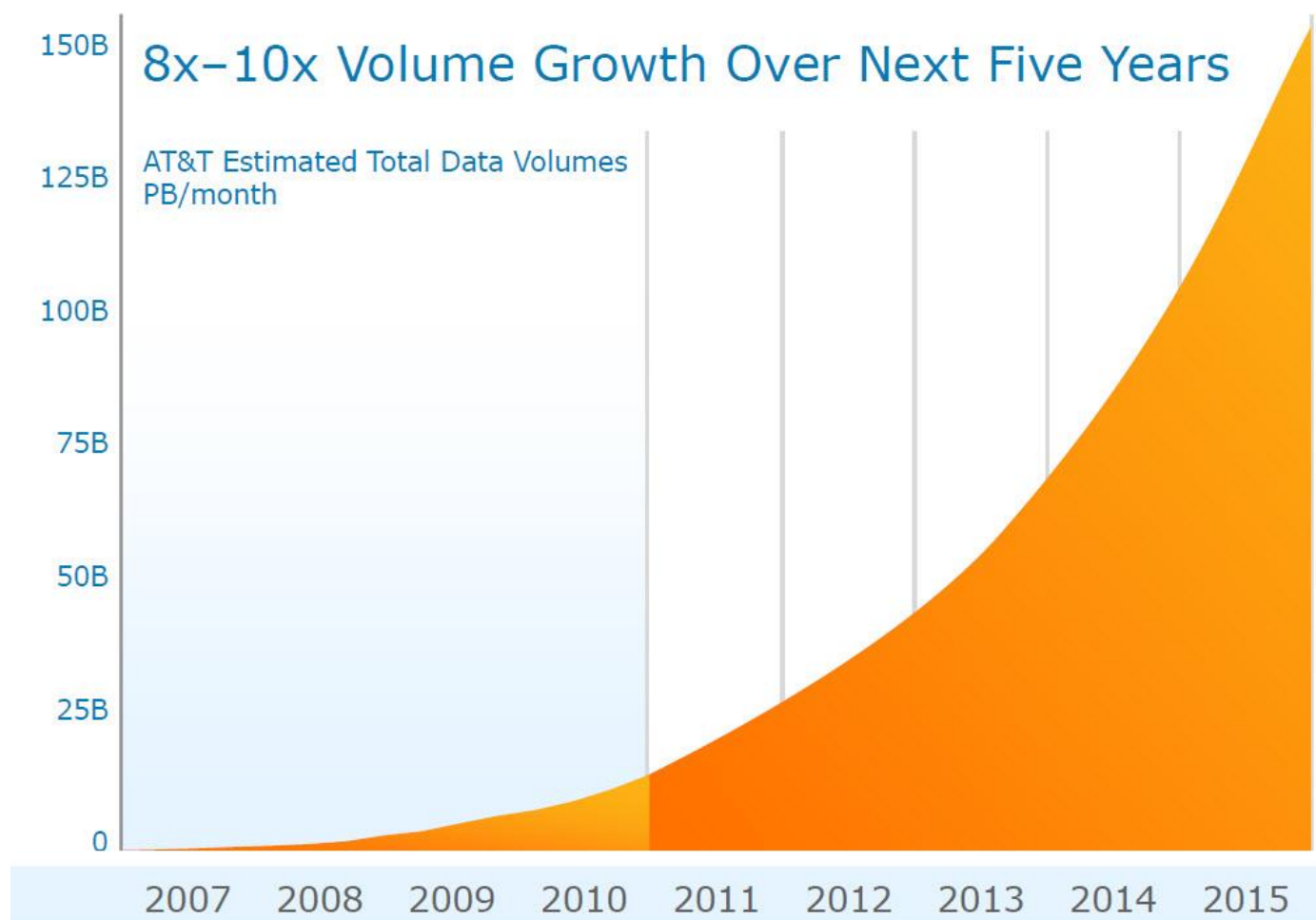


Podaci na Web-u



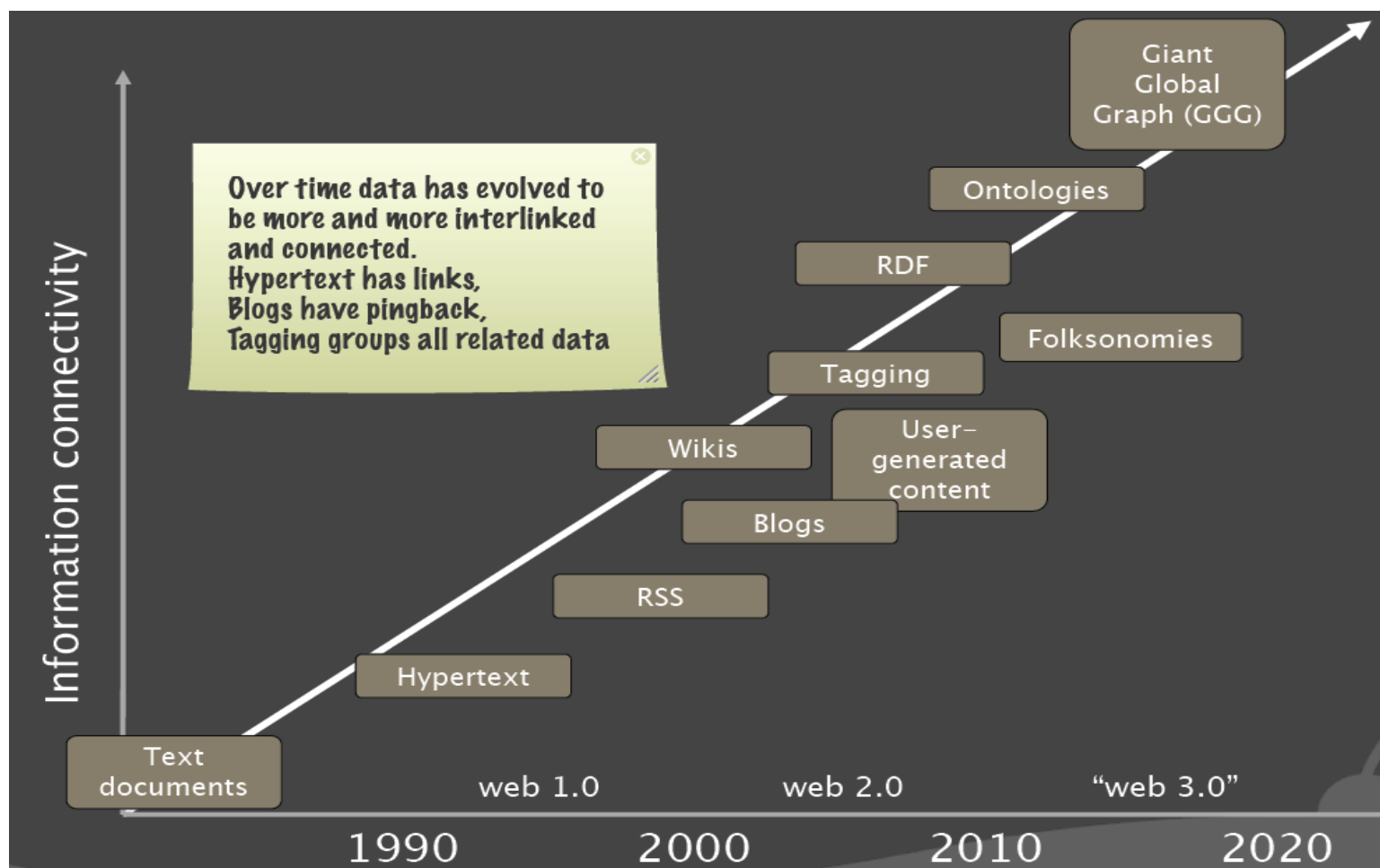


Podaci na Web-u





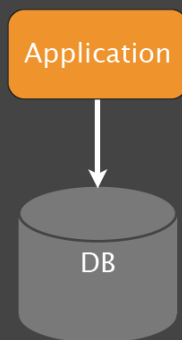
Podaci na Web-u



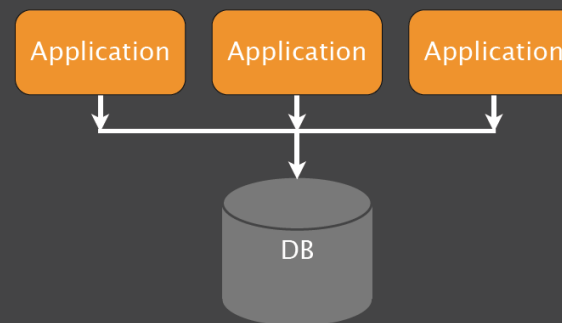


Podaci na Web-u

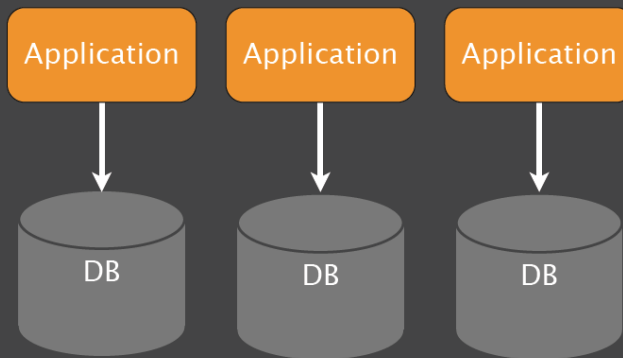
1980s: Mainframe applications



1990s: Database as integration hub



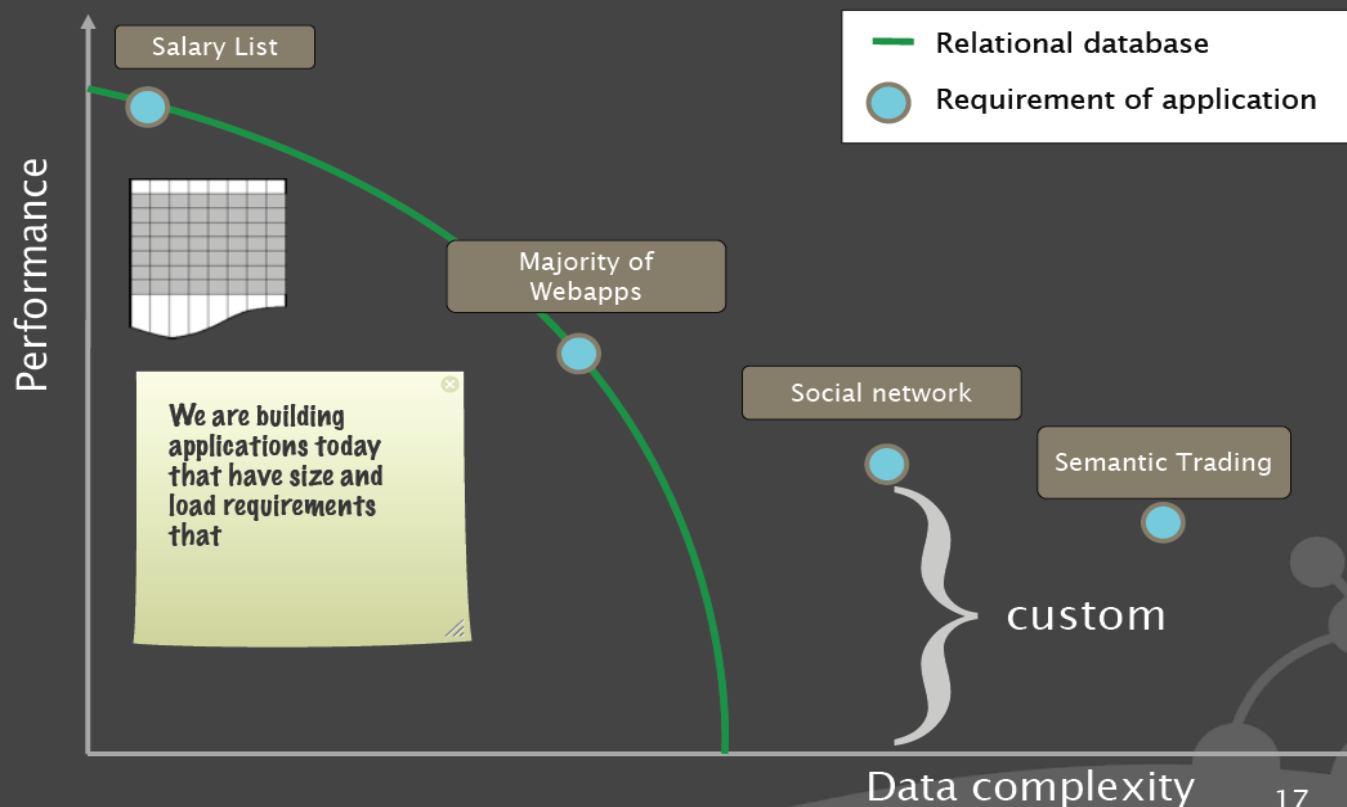
2000s: (moving towards) Decoupled services with their own backend





Podaci na Web-u

RDBMS performance





Podaci na Web-u

- Web aplikacije imaju drugačije potrebe u odnosu na aplikacije za koje su RDBMS razvijane.
- Web aplikacije zahtevaju:
 - Ekstremno veliki broj transakcija u jedinici vremena
 - Dinamička analiza velikih količina podataka
 - Kratko i predvidivo vreme odziva (latency)
 - Skalabilnost (po niskoj ceni)
 - Visok nivo dostupnosti (high availability)
 - Fleksibilnu šemu / polustrukturirane podatke
 - Geografska distribuiranost (veći broj čvorova u kojima se podaci obrađuju, mreža kao problem)



Podaci na Web-u

- Web aplikacijama nisu neophodne:
 - Transakcije
 - Kompleksni SQL upiti
 - Stroga konzistentost
 - Integritet podataka
- Nedostaci RDBMS
 - ACID transakcije nisu skalabilne
 - Horizontalno particionisanje
 - Neefikasni spojevi
 - Transakcije zahtevaju nepotrebnu obradu, odnosno unose dodatni “overhead”
 - Šema relacionih baza podataka nije fleksibilna

Skalabilnost

- **Scaling Up**

- **Dodavanje resursa jedinom čvoru u sistemu**
 - Dodavanje CPU ili memorije
- **Migracija sistema na jaču platformu**
- **Prednosti:**
 - Brzo i jednostavno
- **Nedostaci:**
 - Kada se prevaziđu kapaciteti najačeg sistema
 - Cena
 - Zavisnost od samo jednog proizvođača



Skalabilnost

- **Scaling Out**

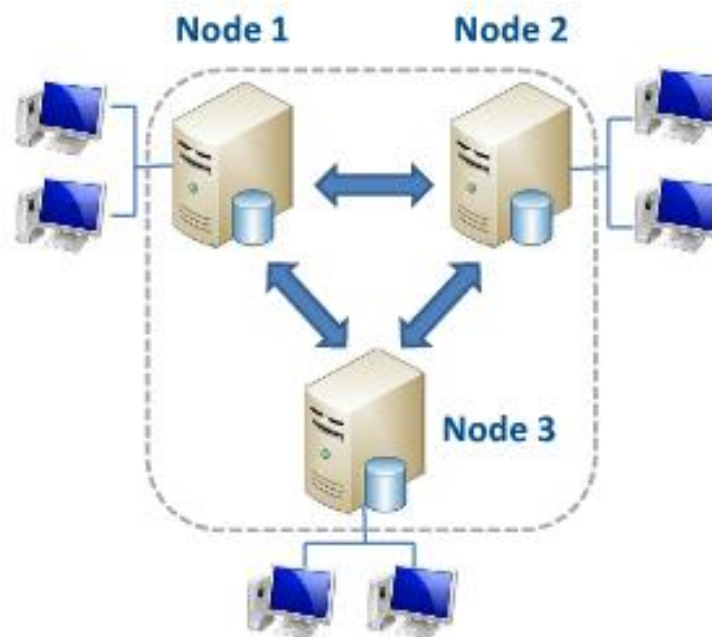
- **Dodavanje novih čvorova u sistem**
- **Funkcionalana (vertikalna) skalabilnost**
 - Grupisanje podataka po funkciji i distribuiranje funkcionalnih grupa u različitim bazama
- **Horizontalna skalabilnost**
 - Distribuiranje istih funkcionalnih grupa u različitim bazama
- **Prednosti:** fleksibilnost
- **Nedostaci:** kompleksnost





Distribuirane baze podataka

- Više čvorova
- Jedna baza podataka





Distribuirane baze podataka

- Zahtevi koje moraju da ispune distribuirane baze podataka:
 - **C**onsistency – sistem se nalazi u konzistentnom stanju posle svake operacije
 - Svi klijenti vide iste podatke
 - **A**vailability – sistem je uvek dostupan (“*always on*”)
 - “*no downtime*”
 - Tolerancija na otkaz čvorova – klijenti uvek imaju pristup nekoj od kopija (replika)
 - Tolerancija na HW/SW promene
 - **P**artition tolerance – sistem funkcioniše čak i u slučaju da ne postoji konekcija između distribuiranih podskupova (pad mreže)
 - Ne samo za čitanje već i za upis



Distribuirane baze podataka

- **CAP Teorema** (E. Brewer, N. Lynch)
 - **U potpunosti** je moguće zadovoljiti **samo 2 od 3** zahteva.
 - Kompromis oko trećeg zahteva
 - **Odustaje se od pristupa “sve ili ništa”**
 - Biraju se različiti nivo konzistentnosti, dostupnosti ili particionisanja.
 - Treba prepoznati koja su od CAP pravila neophodna za funkcionisanje sistema.



Distribuirane baze podataka

- **CA: Consistency & Availability**
 - Kompromis oko Partition Tolerance
 - Karakteristična za single-site cluster rešenja (lakše je obezbediti da su svi čvorovi u stalnom kontaktu)
 - Kada dođe do narušavanja topologije mreže, odnosno do particionisanja mreže, sistem se blokira.
 - Primer: dvofazni komit (2PC)



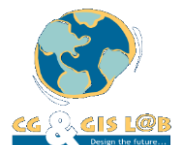
Distribuirane baze podataka

- **CP: Consistency & Partitioning**
 - Kompromis za Availability
 - Pristup pojedinim podacima može biti privremeno onemogućen ili ograničen
 - Ostatak sistema se nalazu u konzistentom/tačnom stanju
 - Primer: horizontalno particionisane baze podataka na većem broju servera (*sharded databases*)



Distribuirane baze podataka

- **AP: Availability & Partitioning**
 - Kompromis za Consistency
 - Sistem je dostupan i prilikom narušavanja mrežne topologije
 - Neki od podataka koje sistem vraća mogu biti privremeno neažurni (***temporarily not up-to-date***)
 - Zahteva strategiju za rešavanje konflikta (***conflict resolution strategy***)
 - Primer: DNS, keš, master/slave replikacija



Distribuirane baze podataka

- **CAE trade-off (Amazon)**
 - **C**ost-efficiency
 - High **A**vailability
 - **E**lasticity
- Biraju se bilo koja dva (C, A, E)
 - Klijent čeka kada je sistem opterećen (C i E)
 - Ukoliko je moguće predvideti opterećenje, moguće je obezbediti A i C rezervisanjem resursa unapred
 - Nepotrebni resursi (over-provisioning) – A i E
- Svi žele A, problem je obezbediti C



Distribuirane baze podataka

- **ACID**

- **A**tomicity

- Kada se deo transakcije ne izvrši, ne izvrši se čitava transakcija. Baza podataka se ne menja.

- **C**onsistency

- Transakcija prevodi DB iz jednog konzistentnog stanja u drugo.

- **I**solation

- Transakcija ne vidi nekomitovane izmene iz drugih transakcija

- **D**urability

- Komitovane izmene u DB su trajne.



Distribuirane baze podataka

- **BASE**

- CAP varijanta ACID svojstava
- **B**asically **A**vailable
- **S**oft **S**tate
- **E**ventually **C**onsistent
- ACID forsira konzistentnost podataka dok BASE prihvata da će se konflikti desiti.



NoSQL baze podataka

- NoSQL baze podataka predstavljaju pokret a ne specifikaciju.
- Prvi put upotrebljen 1998. godine.
- **NoSQL != No SQL**
- **NoSQL == Not Only SQL**
- Termin se upotrebljava za sve nerelacione baze podataka (non-RDBMS)



NoSQL baze podataka

- Internet
- Google
- Bigtable whitepaper (Google) – 2006
- Dynamo whitepaper (Amazon) – 2007
- Cassandra release (Facebook) – 2008
- Voldemort release (LinkedIn) - 2009



NoSQL baze podataka

- Tipična primena:
 - Velike količine podataka (Massive data volumes)
 - Za skladištenje podataka se koristi distribuirana arhitektura
 - Google, Amazon, Facebook – 10K-100K servera
 - Veliki broj upita (Extreme query workload)
 - Nemogućnost efikasnog izvršavanja spojeva kod RDBMS u takvom okruženju
 - Schema evolution
 - Nije jednostavno obezbediti fleksibilnost šeme
 - Promene u šemi se mogu postepeno uvoditi kod NoSQL



NoSQL baze podataka

- Dobre strane:
 - Fleksibilnost
 - Skalabilnost
 - Jednostavne za korišćenje i u pravljanje
 - Eventually consistent
 - Jeftine
 - Prilagođene potrebama Web aplikacija

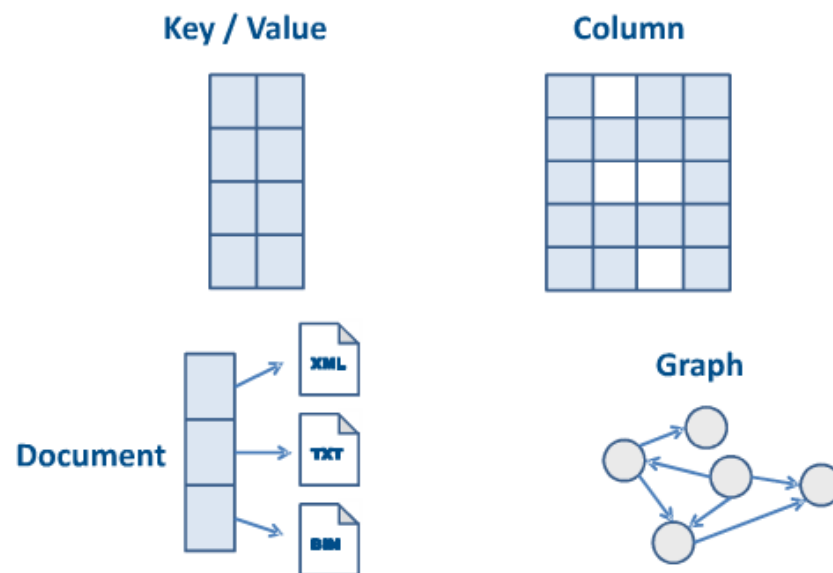


NoSQL baze podataka

- Loše strane:
 - Tehnologija još uvek nije stabilna
 - Ne postoje zajednički standardi
 - Loša podrška za transakcije
 - Loša podrška za pretraživanje podataka
 - Zahteva promenu načina razmišljanja
 - Vrlo je teško naći dva identična scenarija primene.

Taksonomija

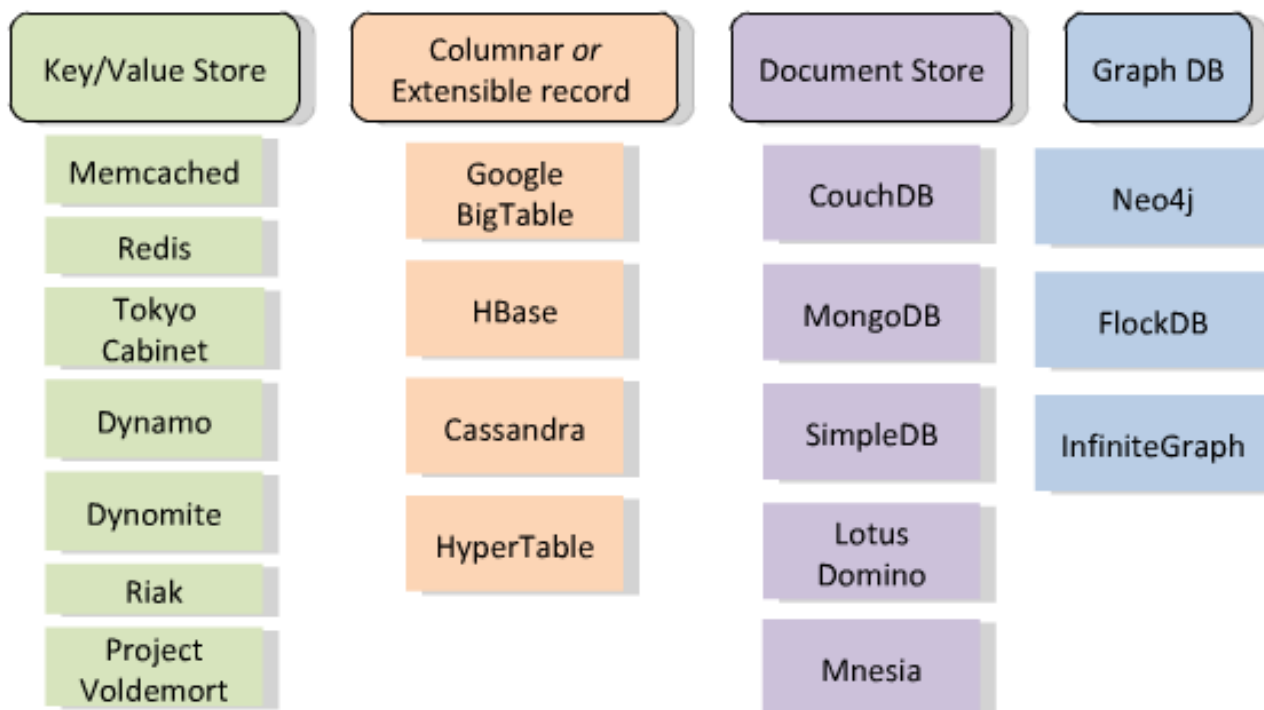
- Key/Value stores
- Column stores (Extensible records)
- Document stores
- Graph databases





Taksonomija

Recent NOSQL database products



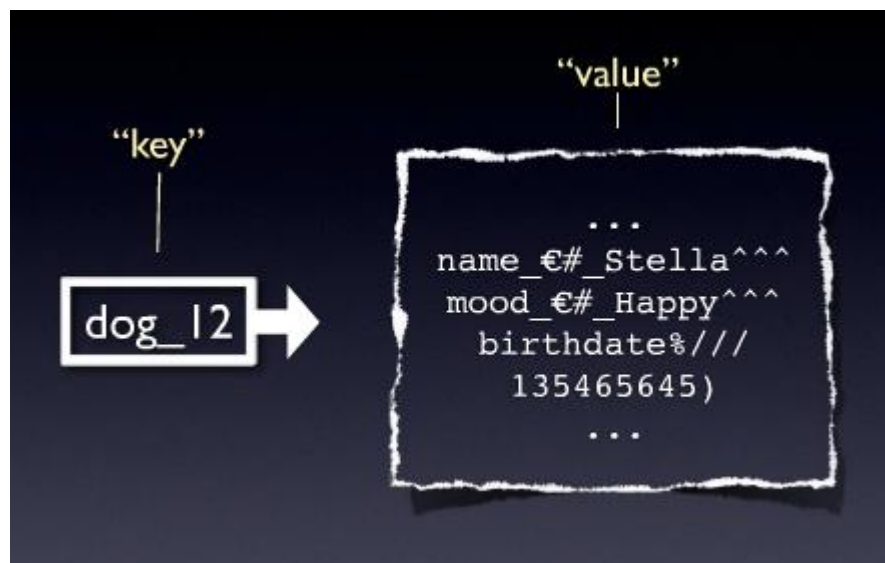


Taksonomija

- **Key/Value stores**
 - Key/Value lookups (DHT), Hash
 - Jedna vrednost, jedan ključ, nema duplikata, izuzetno brzo
 - Skaliranje ogromnih količina podataka
 - Projektovane da podnesu velika opterećenja
 - Podataka je obično BLOB, DB ne razume strukturu podatka.
 - Primer: Riak, Redis, Project Voldemort (Amazon Dynamo whitepaper)

Taksonomija

- **Key/Value stores**



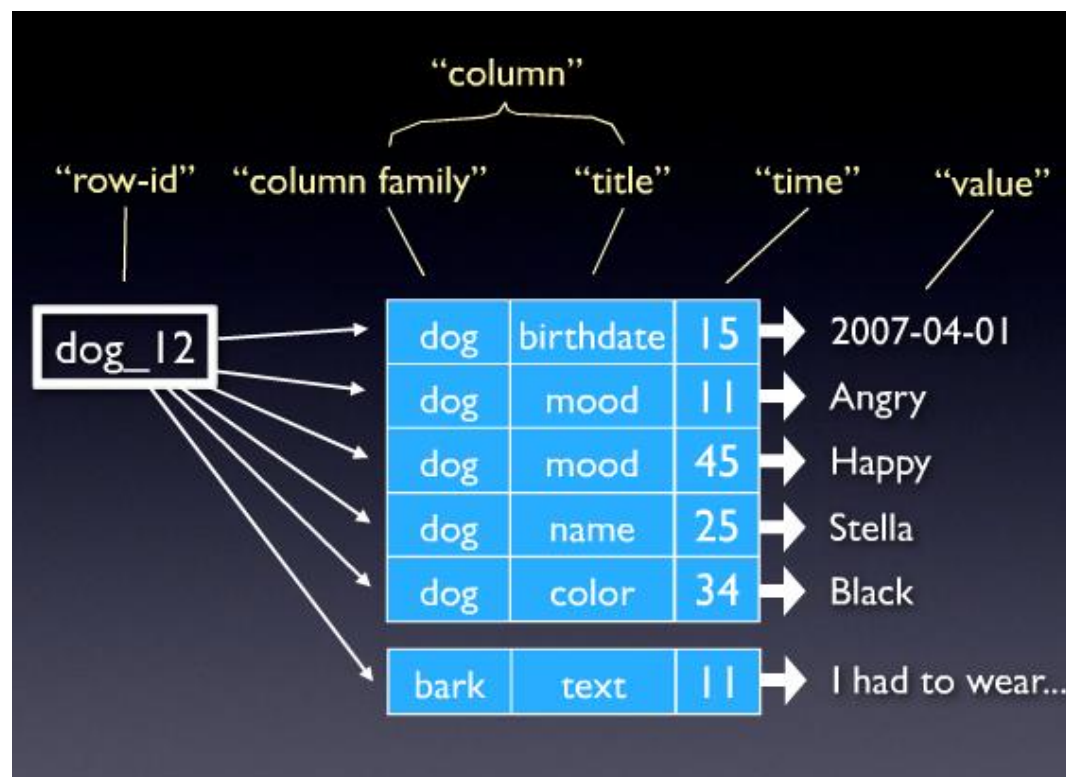


Taksonomija

- **Column stores**
 - BigTable kolonovi
 - Rasuta, distribirana multi-dimenzionalan sortirana mapa
 - Konceptualno:
 - Jedna tabela, beskonačno velika
 - Svaka vrsta može imati različite kolone (po broju i tipu)
 - Tabela je retko posednuta: $|rows| * |columns| > |values|$
 - Primer: Hbase, Cassandra, Hypertable

Taksonomija

- Column stores



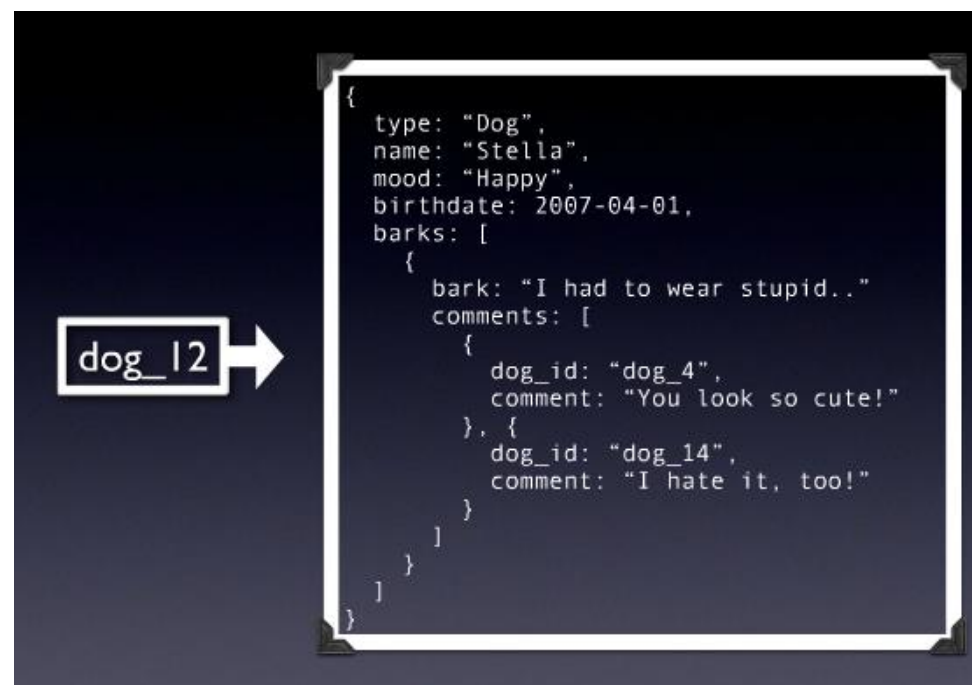


Taksonomija

- **Document stores**
 - Key/Value store, value predstavlja polu-strukturirani dokument čija je struktura razumljiva DB
 - Podaci se mogu pretraživati ne samo po ključu
 - Polu-strukturirani dokumenti (XML, JSON)
 - Primer: MongoDB, CouchDB, Amazon SimpleDB

Taksonomija

- Document stores





Taksonomija

- **Graph databases**

- Inspirisane matematičkom teorijom grafova: $G = (E, V)$
- Modelira se struktura podataka
- Navigacioni model podataka
- Skalabilnost / kompleksnost podataka
- Model: Key/Value parovi za Potege/Čvorove
- Relacije: Potezi između čvorova
- Primer: Neo4j, AllegroGraph, OWLIM

Taksonomija

- **Graph databases**

