

noSQL DataBases

"Not only SQL"

Prema

NoSQL Databases

<http://www.christof-strauch.de/nosql dbs.pdf>

noSQL DataBases

- NoSQL definicija: Sledeća generacija baza podataka koja se odnosi na sledeća svojstva:
 - **ne-relacione, distribuirane, otvorenog koda i horizontalno skalabine.**
- Originalna namera je **moderne veb baze podataka**
- Pokret je počeo 2009. i brzo raste

noSQL DataBases

- Često i sledeće karakteristike: slobodne sheme (**schema-free**), jednostavna podrška replikaciji, jednostavni API, **BASE** (ne ACID, **eventually consistent**), ogromne količine podataka itd.
- Značajna i rastuća industrijska upotreba
- big data
- real-time veb aplikacije

Motivacija za noSQL DB

- Ambicija NoSQL pokreta: “motivacija za traženje alternativa je potreba da se reše problemi za koje relacione baze nisu odgovarajuće”
- Start-ups
- 1998 – termin se odnosio na relacione baze bez podrške SQL-a
- 2009 – na nerelezione sisteme

Razlozi za NoSQL

- Zaobići nepotrebnu kompleksnost, npr, ACID (Facebook status, ...)
- Visoka propusnost (npr. Google –20 petabajta (10^{15} bajta / pebibyte) podataka dnevno u Bigtable – column store)
- Horizontalna skalabilnost i izvršavanje na pristupačnom hardveru (**Commodity Hardware**)
 - Rešavaju se problemi RDBMS:
 - Skaliranje podataka
 - Performanse centralizovanih servera
 - Rigidno projektovanje sheme

Razlozi za NoSQL (nast.)

- Izbegavanje skupog objektno-relacionog preslikavanja
- Kompromisi između pouzdanosti i boljih performansi
- Koncept “One size fit’s it all” baze podataka bio i jeste pogrešan
- Pokreti u programskim jezicima i razvojnim okruženjima

Potrebe

- danas vs. juče
 - Predviđeni padovi sistema/ nema padova
 - Statički – predefinisani upiti / dinamički
 - Slobodna struktura podataka / Rigidna struktura podataka
 - Distribuirana / centralizovana instalacija

One Size Fits All: An Idea Whose Time has Come and Gone

Michael Stonebraker, CACM, 2008.

- osnovna arhitektura RSUBP koju prodaju komercijalni prodavci je starija od dve decenije
- u međuvremenu računari na kojima su SUBP su dramatično napredovali
- gridovi (mreža, grids; slabo povezani heterogeni, geografski raspršeni sistemi) su zamenili deljenu memoriju multiprocesora
- brzina CPU je značajno porasla
- unutrašnja memorija je mnogo brža i veća
- diskovi su mnogo veći
- nove značajne aplikacije SUBP tehnologije
 - skladišta podataka, polu-strukturani podaci, naučni podaci

One Size Fits All: An Idea Whose Time has Come and Gone

Michael Stonebraker, CACM, 2008.

- Tradicionalna arhitektura RSUBP može značajno da se “potuče” (faktor 25–50) specijalizovanom implementacijom na svakom od značajnijih SUBP tržišta
 - U oblasti *skladištenja podataka (data warehousing)*, reklo bi se da je reč o tzv. ***kolonskom skladištenju***

Skladištenje podataka: *kolonsko skladištenje*

- SUBP mašina (engine) mora da čita samo one elemente podataka koji su relevantni za konkretni upit
- Kompresija podataka je mnogo efektivnija u kolonskom skladištenju zato što se komprimuje samo jedan tip podataka po bloku skladištenja
- Ako SUBP mašina radi nad komprimovanim podacima, onda ima manje kopiranja i bolje je korišćenje L2 keša
- CPU vreme se dramatično redukuje
- 1990 (MonetDB and SybaseIQ) i noviji komercijalni proizvodi firmi Vertica, Infobright i Paracel.

Druga tržišta baza podataka

- Na drugim tržištima baza podataka, uključujući obradu poslovnih podataka, specijalizovane arhitekture nude slične prednosti
- “početak kraja” "one-size-fits-all" sistema koje prodaju najveći SUBP prodavci
- Specijalizovane arhitekture postaće dominantne u nekim oblastima primene SUBP u narednim decenijama – za korisnike koji su zainteresovani za performanse
- Na drugom kraju - sistemi otvorenog koda kao što su MySQL, Postgres, Ingres - privlače pažnju
- Novi SUBP početnici (startups) sa specijalizovanim implementacijama

MapReduce

- Model programiranja, paradigma, za obradu velikih skupova podataka paralelnim distribuiranim algoritmom na klasteru
- Program MapReduce
 - procedura **Map()** koja izvodi filterisanje i sortiranje (npr. sortiranje studenata po imenu u redove, jedan red za svako ime)
 - **Reduce()** operacije sumiranja (brojanje studenata u svakom redu - frekvenicja imena)
- Model inspirisan funkcijama [map](#) i [reduce](#) iz funkcionalnog programiranja

MapReduce

- Biblioteke MapReduce napisane su u više programskih jezika, sa različitim nivoom optimizacije.
- Jedna popularna implementacija otvorenog koda je [Apache Hadoop](#).
- Ime MapReduce originalno se odnosilo na svojstvo [Google](#) tehnologije
- MapReduce paradigma je prisutna u mnogim programskim jezicima (npr. Python), okruženjima (npr. Apache Hadoop), čak i u JavaScript alatima (npr. Dojo) i NoSQL bazama podataka (npr. CouchDB).
- Zato što se dobro uklapa u distribuiranu obradu

Primeri NoSQL

- Cassandra originalno razvijena za Fejsbuk a danas se koristi na Tviteru i Digg (agregator vesti)
- Projekat Voldemort je razvijen i koristi se na LinkedIn,
- servisi u oblaku kao što je NoSQL Amazon SimpleDB i Ubuntu One, skladište u oblaku i servis za sinhronizaciju zasnovan na CouchDB
- Ovi korisnici NoSQL skladišta podataka su po svojoj prirodi veoma zainteresovani za dalji razvoj nerelecionih rešenja koja koriste

Primeri NoSQL

- Ipak, većina popularnih NoSQL skladišta podataka je usvojila ideje ili od
- ***Google Bigtable ili***
- ***Amazonovog Dynamo***
- Bigtable-inspirisana NoSQL skladišta obično se označavaju kao ***kolonska skladišta*** (npr. HyperTable, HBase)
- Dynamo je uticao na većinu ***ključ-vrednost*** skladišta (npr. Cassandra, Project Voldemort)

CAP teorema

- Eric Brewer, “Towards Robust Distributed Systems”, 2000.
- **Consistency (C)**: if and how a system is in a consistent state after the execution of an operation
- **Availability (A)**: allows it to continue operation (i. e. allowing read and write operations) if e. g. nodes in a cluster crash or some hardware or software parts are down due to upgrades
- **Partition Tolerance (P)**: ability of the system to continue operation in the presence of network partitions
- **T: 2/3**
- **ACID – CP**
- **BASE – AP** - Amazon’s Dynamo
- Google’s Bigtable - CA

NoSQL baze podataka- klasifikacija prema modelu podataka

- Skladišta ključ, vrednost
- Skladištenje dokumenata (Document Store)
- Kolonska skladišta / Kolonske familije
- Grafovske baze podataka
- Višemodelne baze podataka (Multimodel Databases)
- Objektne baze podataka
- Baze podataka na mreži i u oblaku (cloud)(Grid & Cloud Database Solutions)
- XML baze podataka
- Višedimenzione baze podataka
- Viševrednosne baze podataka (Multivalue Databases)

Skladišta ključ/vrednost

- Jednostavni model podataka : mapa/rečnik
- Dopušta upis i čitanje vrednosti po ključu
- **Amazon's Dynamo**
- Dinamički upiti, kompleksni (SQL)– nepotrebni
- Velika količina podataka
- Skalabilnost – jednostavnost dodavanja uređaja/servera
- Dopušta programerima da skladište podatke bez šeme bp; podatak se obično sastoji od stringa (ključ) i vrednosti
- Particionisanje/replikacija/verzije/B-stabla
- Konkurentna azuriranja...; matematika
- Ne podržava intervalne upite ili batch operacije

Skladišta ključ/vrednost

- Na primer,

user1923_color Red

user1923_age 18

user3371_color Blue

user4344_color Brackish

user1923_height 6' 0"

user3371_age 34

Skladišta ključ/vrednost

- Jedna masivna “tabela” na distribuiranom čvoru, na primer:
 - user1923_color Red
 - user1923_age 18
 - user4344_color Brackish
 - user1923_height 6' 0"
 - user3371_age 34
 - error_msg_457 There is no file %1 here
 - error_message_1 There is no user with %1 name
 - 1923_name Jim
 - user1923_name Jim Smith
 - user1923_lname Smith
 - Application_Installed true
 - log_errors 1
 - install_path C:\Windows\System32\Restricted
 - ServerName localhost

Amazon's Dynamo

- Infrastruktura se sastoji od desetina hiljada servera i mrežnih komponenti lociranih u mnogim centrima podataka širom sveta
- Koristi se pristupačni hardver (commodity hardware)
- Pad (otkaz) komponenti je “standardni način rada”
- Koristi krajnje decentralizovanu arhitekturu, slabo povezanu, servis-orijentisanu
- **get() i put() operacije**

Skladište dokumenata

- Centralni koncept skladišta dokumenata je "dokument".
- Dokument kodira podatke (ili informacije) u nekom standardnom formatu ili kodu
- Npr. XML, YAML (ključ-vrednost), JSON (JavaScript Object Notation; atribut-vrednost)
- binarni oblici BSON (u MongoDB), PDF i Microsoft Office dokumenti (MS Word, i sl.)

Skladište dokumenata

- Sledeći korak u odnosu na jednostavna ključ-vrednost skladišta ka nešto složenijim i smislenijim podacima
- Nema striktne sheme
- Model podataka dokumenta: svaki slog sa pridruženim podacima je “dokument”

Skladište dokumenata: Apache CouchDB

- CouchDB je baza poataka dokumenata napisana u Erlang (declarative, functional language for concurrent, distributed systems)
- “Cluster of unreliable commodity hardware”
- Schema-free dokumenta
- JavaScript upitni jezik
- Distribuirana, replikacija između servera i servera i klijenata
- Korisnici: *ubuntu one* – servis za skladištenje i replikaciju u oblaku Ubuntu Linux
- CouchDB je deo BBC nove platforme veb aplikacija

Apache CouchDB (nast.)

- Polu-strukturirana baza podataka
- Dokument – glavna struktura i apstrakcija
- Sastoji se od imenovanih polja sa parovima (ključ, vrednost)
- Primer: wiki članak
 - " Title " : " CouchDB ",
 - " Last editor " : "172.5.123.91" ,
 - " Last modified " : "9/23/2010" ,
 - " Categories " : [" Database " , " NoSQL " , " Document Database "],
 - " Body " : " CouchDB is a ... " ,
 - " Reviewed " : false
- Indeksiranje dokumenata se vrši B-stabloma po (ID, revision no) dokumenta
- Pogledi, verzije, replikacija i distribucija

Mongo DB

- Schema-free baza podataka dokumenata napisana u C++
- Glavni cilj: popuniti jaz između brzih i jednostavnih ključ/vrednost skladišta i bogatih RDBMS
 - New York Times
 - Distribuirana društvena mreža DIASPORA*
- Dokument – apstrakcija i jedinica podataka
- Struktura uporediva sa XML dokumentom, Python rečnikom, Ruby heš (asocijativni niz) ili JSON (JavaScript Object Notation) dokumentom
- BSON format

Dokument u JSON notaciji

```
{  
title : " MongoDB ",  
last_editor : "172.5.123.91" ,  
last_modified : new Date ("9/23/2010") ,  
body : " MongoDB is a..." ,  
categories : [" Database ", " NoSQL ", " Document  
Database "],  
reviewed : false  
}
```

Kreiranje, dodavanje u MongoDB kolekciju, pretraživanje i ažuriranje:

```
db.createCollection (<name >, {< configuration parameters >})
db.< collection >.insert ( { title : " MongoDB ", last_editor : ... } );
db.< collection >.find ( { categories : [ " NoSQL ", " Document
    Databases " ] } );
db.< collection >.save ( { ... } );
db.< collection >.update ( { title : " MongoDB "}, { $inc : { revision : 1}}
    );
```

Indeksno B-stablo

```
db.< collection >.ensureIndex ({< field1 >:< sorting >, <field2 >:<
    sorting >, ...});
```

MongoDB: Geospatial Indexes

- Dvodimenziono geoprostorno indeksiranje
- Koordinate:
- `{ loc : { 50, 30 } }` // koordinate u polju niza
- `{ loc : { x : 50, y : 30 } }` // koordinate u polju objekta
- `db.< collection >. createIndex ({ <field > : "2d" });`
- Upiti vezani za lokaciju, npr. “naći n objekata najbližih specifičnoj lokaciji”

Kolonski orijentisane baze podataka

- Pristup skladištenju i obradi podataka po kolonama umesto po vrstama – poreklo u analitici i poslovnoj inteligenciji
- Obrada po kolonama, bez deljenja podataka u arhitekturi masovno paralelne obradje može da doprinese visokoj performantnosti aplikacija
- Proizvodi: Sybase IQ i Vertica
- Google's Bigtable
- Cassandra, inspirisana Bigtable i Amazon's Dynamo.

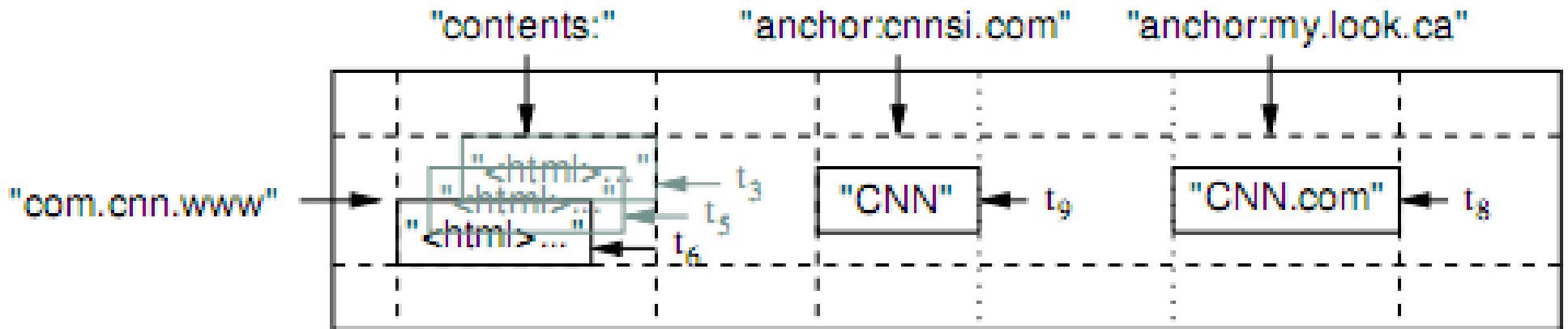
Google's Bigtable

- Bigtable se opisuje kao “distribuirani sistem skladištenja za upravljanje strukturnim podacima vrlo velikih obima”: petabajti podataka na hiljadama pristupačnih (“commodity”) servera
- Koristi ih preko 60 projekata u Google (podatak iz 2006.) – za indeksiranje veba, Google Earth, Google Analytics, Orkut (Google društvena mreža) i Google Docs

Bigtable model podataka

- Bogatiji od jednostavnog “ključ/vrednost” modela
- Podržava polustrukturirane podatke
- Struktura podataka: “distribuirana, perzistentna, višedimenziona sortirana mapa”
- Adresiraju se trojkom (ključ-vrste, ključ-kolone, vremenska-oznaka)

Primer: rezultati veb popisivača (crawler) u Bigtable



- Mapa: nefiksirani broj vrsta koje predstavljaju domene koje je crawler pročitao
- Nefiksirani broj kolona
 - Prva: contents – sadržaj stranica
 - Ostale: anchor:<domain-name> - tekst linkova iz referišućih domena – svaki se predstavlja posebnom kolonom
 - Svaka vrednost ima pridruženu vremensku oznaku

Bigtable

- Vrste se čuvaju u leksikografskom poretku i dinamički particionišu u tzv. tablete (*tablets*)
 - “the unit of distribution and load balancing”
- Broj kolona po tabeli nije ograničen
- Kolone su grupisane po prefiksu ključa u skupove - *familije kolona*
- Prethodni primer: dve kolonske familije - *content* i *anchor*
- Vremenske oznake, predstavljene 64-bitnim int

Apache Cassandra

- Amazon's Dynamo i Google's Bigtable
- Razvijena za Fejsbuk, open-sourced 2008.
- Obrada velikih, brzorastućih količina podataka (100 miliona korisnika u junu 2008, 250 miliona u avgustu 2009 i preko 1.1 milijarde korisnika u martu 2013)
- Visoka i inkrementalna skalabilnost
- Isplativost
- Tretman otkaza kao “pravilo pre nego izuzetak”
- ...

Cassandra: model podataka

- Instanca se obično sastoji od jedne tabele - “distribuirana višedimenziona mapa indeksirana ključem”
- Tabela je strukturirana po sledećim dimenzijama:
 - Vrsta identifikovana string-ključem proizvoljne dužine
 - Kolonska familija – u proizvoljnom broju po vrsti
 - Kolone imaju ime i pamte veći broj vrednosti po vrsti, koje se idenifikuju vremenskom oznakom
 - Superkolone imaju ime i proizvoljni broj pridruženih kolona
- Vrednosti se adresiraju trojkom (ključ-vrste, ključ-kolone, vremenska-oznaka)
- Ključ-kolone:
 - familija-kolona:kolona (za jednostavne kolone) ili
 - familija-kolona:superkolona:kolona (za kolone u okviru superkolone)

Cassandra: API

- Klijent-aplikacija definiše “kvorum” replika za čitanje i ažuriranje - stepen “eventual consistency”
- `get(table, key, columnName)`
- `insert(table, key, rowMutation)`
 - (semantika ista kao za `update`)
- `delete(table, key, columnName)`

List Of NoSQL Databases [currently 150]Core NoSQL Systems: <http://nosql-database.org/>

- **Column Store / Column Families**

- [Hadoop / Hbase](#)
- [Accumulo](#)
- [Cassandra](#) – Facebook, sada deo Apache Software Project.
- [Hypertable](#)
- [Amazon SimpleDB](#)
- [Cloudata](#)
- [Cloudera](#)
- [HPCC](#)
- [Stratosphere](#)

Document Store

- [MongoDB](#)
- [Couchbase](#)
- [Elasticsearch](#)
- [CouchDB](#)
- [RethinkDB](#)
- [RavenDB](#)
- [MarkLogic Server](#)
- [Clusterpoint Server](#)
- [ThruDB](#)
- [Terrastore](#)

Key Value / Tuple Store

- [DynamoDB](#) (Amazon)
- [Riak](#)
- [Redis](#)
- [Voldemort](#)
- [Azure Table Storage](#)
- [Aerospike](#)
- [FoundationDB](#)
- [LevelDB](#)
- [Berkeley DB](#)
- [GenieDB](#)

Grafovske baze podataka

- Za podatke čiji se odnosi pogodno predstavljaju grafovima (elementi međusobno povezani neodređenim brojem veza)
- Društvene veze, veze javnog prevoza, putne mape, mrežne topologije, i sl.
- **Neo4J**
- **AllegroGraph**
- IBM DB2 (jezik SPARQL, od v. 10(
- **Bigdata**
- **Infinite Graph**
- **InfoGrid**
- **HyperGraphDB**
- **DEX**

Višemodelne baze podataka

- Koriste fleksibilnu kombinaciju ključ-vrednost parova, dokumenata i grafova
- Može da ponudi raznovrsnost logičkih modela, ili pogleda na uskladištene podatke
- [ArangoDB](#)
- [OrientDB](#)
- [Datomic](#)
- [FatDB](#)
- [AlchemyDB](#)

Objektne baze podataka

- [Versant](#)
- [Db4o](#)
- [Objectivity](#)
- [Starcounter](#)
- [Perst](#)
- [VelocityDB](#)
- [HSS Database](#)
- [ZODB](#)
- [Magma](#)
- [NEO](#)

Baze podataka nad mrežom i u oblaku (Grid & Cloud Database Solutions)

- Od 1990, mrežno izračunavanje se koristi kao revolucionarna paradigma pristupa i upravljanja distribuiranim, heterogenim i geografski raspršenim resursima, sa pristupom jednostavnim kao elektromreži
- Resursi podrazumevaju i baze podataka
- [GigaSpaces](#)
- [GemFire](#)
- [Infinispan](#)
- [Queplix](#)
- [Hazelcast](#)

XML baze podataka

- [EMC Documentum xDB](#)
- [eXist](#)
- [Sedna](#)
- [BaseX](#)
- [Qizx](#)
- [Berkeley DB XML](#)

Multidimenzione baze podataka

- Online analytical processing (OLAP) je višedimenziono, višekorisničko, klijent-server računarsko okruženje za korisnike koji treba da analiziraju kompanijske podatke
- Ključne za OLAP sisteme jesu višedimenzione baze podataka koje ne samo da konsoliduju i proračunavaju podatke već obezbeđuju pretragu i izračunavanje raznovrsnih podskupova podataka.
- Višedimenziona baza podataka podržava višestruke poglede na skupove podataka za korisnike koji treba da analiziraju odnose među kategorijama podataka
- [Intersystems Cache](#)
- [GT.M](#)
- [SciDB](#)
- [MiniM DB](#)
- [rasdaman](#)

Viševrednosne baze podataka

- **MultiValue je tip [NoSQL](#)** i višedimenzione baze podataka
- Razlikuju se od relacionih u tome što imaju svojstva koja podržavaju i podstiču korišćenje atributa koji mogu da uzmu listu vrednosti umesto da budu single-valued
- Može im se pristupiti SQL-om ili bez njega
- [U2](#)
- [OpenInsight](#)
- [TigerLogic PICK](#)
- [Reality](#)
- [OpenQM](#)
- [Model 204 Database](#)
- [ESENT](#)
- [jBASE](#)

Why Enterprises Are Uninterested in NoSQL

By Michael Stonebraker

September 30, 2010

- According to [a recent ReadWriteWeb blog post](#) by Audrey Watters, 44% of enterprise users questioned had never heard of [NoSQL](#) and an additional 17% had no interest. So why are 61% of enterprise users either ignorant about or uninterested in NoSQL?
- a very senior technical guru at a large enterprise
- He then made one comment about OLTP, one comment about warehouses, and one general comment. These follow.
- **No ACID Equals No Interest**
- **A Low-Level Query Language is Death**
- **NoSQL Means No Standards**
 - there are many different NoSQL engines, each with a different user interface and a different model which is unique to that system.