NJË STUDIM KRAHASUES I SISTEMEVE NoSQL PËR MENAXHIMIN E BIG DATA

***BRISILDA MUNGULI.[1], REZARTA KAPAJ.[2]**

Universiteti i Tiranës, Fakulteti i Shkencave të Natyrës, Departamenti i Informatikës

e-mail: brisilda.munguli@fshn.edu.al

Përmbledhje

Zhvillimi i teknologjive të reja web ka sjell sfida të reja. Bota dixhitale po rritet me shpejtësi të madhe duke prodhuar qindra terabyte dhe petabyte të dhënash të shumëllojshme, të cilat mund të jenë të strukturuara, gjysëm të strukturuara ose të pastrukturuara. Ky njihet si fenomeni global i Big Data i cili i referohet grupeve të të dhënave me volum të madh dhe komplekse të cilat janë të vështira për t'u menaxhuar nga sistemet tradicionale të bazave të të dhënave. Për ti bërë ballë këtij fenomeni janë zhvilluar aplikacione dhe koncepte të reja për bazat e të dhënave. Bazat e të dhënave tradicionale përdoren kryesisht për menaxhimin e të dhënave të strukturuara dhe vështirë të shkallëzohen me madhësinë në rritje të grupeve të të dhënave. Zgjidhjet NoSQL në thelb kanë për qëllim të zgjidhin problemet e Big Data që bazat e të dhënave relacionale nuk janë të përshtatëshme, janë të kushtueshme për t'u përdorur ose kërkojnë zbatimin e praktikave që cënojnë natyrën relacionale të tyre. Në këtë artikull ne prezantojmë një studim teorik të disa prej sistemeve të bazave të të dhënave NoSQL më të përdorëshme duke vënë në dukje karakteristikat që i bëjnë ato një alternativë të mirë kundrejt RDBMS dhe duke përshkruar qëllimin dhe funksionalitet e tyre në mënyrë që të jemi në gjendje të përcaktojmë se cila është më e përshtatëshme për të plotësuar nevojat e aplikacionit tonë.

Fjalëkyçe: NoSQL, Big Data, RDBMS.

Abstract

The emerging of new web technologies have brought new challenges. Digital world is growing with velocity producing hundred of terabyte to petabyte data which vary and can be structured, semi- structured or unstructured. This is known as the global phenomenon of Big Data which refers to data sets so large and complex that are impractical to manage with traditional database systems. New applications and concepts of database such as NoSQL databases has been evolved to cope with this phenomenon. The traditional RDBMS are mainly for management of structured data and hard to scale out to the growing size of the data sets. NoSQL Solutions are basically meant to solve a big data problem that relational databases are either not well suited for, too expensive to use or require you to implement something that breaks the relational nature of your DB. In this paper we represent a theoretical study of various popular NoSQL databases by pointing out the features that make them a competitive alternative to RDBMS for Big Data Management and describing their purposes and functionality so that we can decide which one better suits to our application's need.

Key words: NoSQL, Big Data, RDBMS.

1. Hyrje

Zgjidhjet NoSQL në thelb kanë për qëllim të zgjidhin problemet e Big Data që bazat e të dhënave relacionale nuk janë të përshtatëshme, janë të kushtueshme për t'u përdorur ose kërkojnë zbatimin e praktikave që cënojnë natyrën relacionale të tyre. Bazat e të dhënave NoSQL sigurojnë shpejtësinë, shkallëzimin dhe performancën që duhet për të punuar me grupe të dhënash masive për nxjerrjen e të dhënave nga rrjetet sociale, analizimin e të dhënave të sensorëve, analizat e të dhënave të tregjeve dhe skenarë të tjerë që kanë të bëjnë me Big Data. Në këtë studim do njihemi në paragrafin e parë me sfidat që zgjidhin sistemet NoSql dhe karakteristikat e bazave të të dhënave tradicionale që ato sakrifikojnë në favor të performancës dhe përshkallëzmit. Në paragrafin e dytë do të japim një klasifikim të sistemeve NoSql në 4 grupe duke përdorur si kriter modelimin e të dhënave. Nga këto 4 grupe do të fokusohemi në 1 prej tyre- bazat e të dhënave të oreintuara në dokument nga i cili kemi zgjedhur dy sisteme përfaqësues: mongoDB dhe CouchDB të cilat do ti krahasojmë me njëri-tjetrin duke evidentuar karakteristikat që pritet të ketë një sistem NoSQL. Si përfundim do të nxjerrim një konkluzion se cilin prej këtyre sistemeve duhet të zgjedhim për të plotësuar nevojat e aplikacionit jonë.

2. SQL-ACID krahasuar me NoSQL-BASE

Bazat e të dhënave relacionale sigurojnë konsistencë strikte nëpërmjet kufizimeve ACID: Atomiciteti, konsistenca, izolimi dhe vazhdimësia.

Bazat e të dhënave NoSQL heqin dorë nga kufizimet ACID në favor të:

- Shkallëzimit
- Një performance më të mirë

Analog me termin ACID për bazat e të dhënave NoSQl përdoret termi BASE: Një aplikacion që është i disponueshëm gjatë gjithë kohës, mund të mos jetë gjatë gjithë kohës konsistent, por do të jetë përfundimisht konsistent në një gjendje të caktuar.

2.1. Teorema CAP

Teorema CAP është formuluar nga: E. Brewer. (2000) dhe haset shpesh kur përmenden sistemet NoSQL. Sipas kësaj teoreme një sistem mund të ketë vetëm 2 nga 3 karakteristikat e mëposhtëme: konsistencë, disponueshmëri dhe tolerancë ndaj dëmtimeve.

- Konsistenca: Të gjithë klientët do të shikojnë të njëjtat të dhëna gjatë gjithë kohës.
- Disponueshmëria: Do të ketë gjithmonë një përgjigje për çdo veprim leximi ose shkrimi.

- Toleranca ndaj dëmtimeve: Sistemi do të jetë funksional pavarësisht dëmtimeve në rrjet ose ndonjë prej nyeve të tij.

Sistemet NoSQL zakonisht heqin dorë nga konsistenca në favor të disponueshmërisë dhe tolerancës ndaj dëmtimeve.

3. Tipet e bazave të të dhënave NoSQL

Egzistojnë rreth 225 baza të të dhënave NoSQL (http://nosql-database.org) të listuara në faqen web NoSQL-org.com çka e bën pothuajse të pamundur krahasimin e tyre dhe përzgjedhjen e një sistemi përfaqësues për nevojat tona. Brenda këtij numri kaq të madh databazash mund të bëjmë disa nëndarje duke përdorur si kriter modelimin e të dhënave. Në varësi të mënyrës se si sistemet NoSQL i modelojnë të dhënat ato klasifikohen në 4 grupe:

- Çelës/Vlerë
- Kolonë
- Dokument
- Grafë

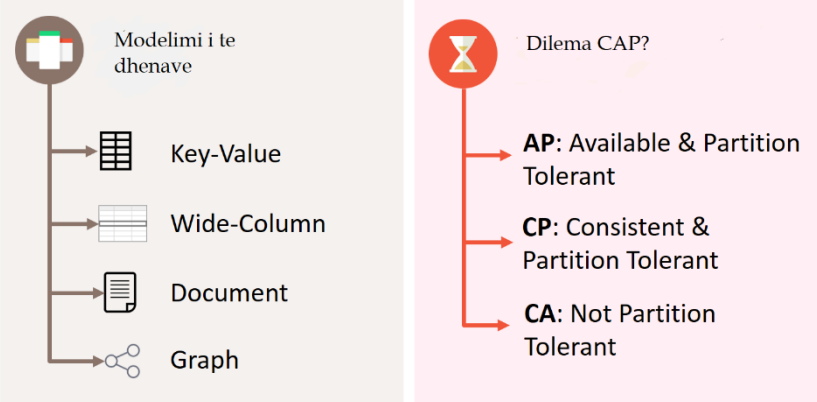

Figura 1. Klasifikimi i sistemeve NoSQL në bazë të modelimit të të dhënave

Brenda këtij grupimi do të studjojmë bazat e të dhënave të orientuara në dokument.

3.1. Modelimi i të dhënave në bazat e të dhënave NoSQL të orientuara në dokument

Modelimi i të dhënave ka të bëjë me mënyrën sesi ruhen dhe aksesohen të dhënat. Bazat e të dhënave të bazuara në dokument i ruajnë të dhënat si dokumenta zakonisht në formatin JSON, ku një dokument është i ngjashëm me objektet në OOP. Një dokument përbëhet nga 1 ose disa fusha ku çdo fushë ka një vlerë të një tipi të caktuar psh: string, datë, binare ose tabelë. Lidhjet e objekteve me njëri-tjetrin nuk ruhen në dokumenta të veçanta por brenda një dokumenti të vetëm duke lehtësuar aksesimin e të dhënave dhe evituar nevojën për të bërë bashkime ndërmjet tabelave.

Bazat e të dhënave NoSql dokument kanë karakteristikat e mëposhtëme:

- Janë schema free- nuk kanë një skemë statike, ajo është e ndryshueshme.
- Çdo dokument idendifikohet nga një id që përcaktohet nga përdoruesi ose gjenerohet automatikisht.
- Një dokument përbëhet nga disa fusha.
- Kërkimi i shpejtë, sepse gjithë lidhjet e të dhënave janë brenda dokumentit.

Figura 2. Baza e të dhënave NoSQL e orientuar në dokument

Bazat e të dhënave dokument, ofrojnë fleksibilitet në aksesin e të dhënave, dokumentat mund të merren jo vetëm nga çelësi por edhe nga vlera të fushave brenda dokumentit.

4. Krahasimi mongoDB, CouchDB

Në këtë paragraf do të krahasojmë bazat e të dhënave NoSQL mongoDB dhe CouchDB.

Figura 3. Krahasimi mongoDB, CouchDB

Ka dy arsye se pse kemi zgjedhur këto dy sisteme përfaqësuese:

- Janë me kod të hapur
- Kanë popullaritet të madh:

○ **MongoDB** përdoret nga: Google, UPS i-parcel, Facebook, Expedia, ebay, SAP, Nokia, Verizon, Forbes, ThermoFisher, Amadeus etj.(CP)

○ **CouchDB** përdoret nga: ebay, doddle, centeredge, cisco, sky, matrix, Tommy Hilfiger, Ryanair, Viber, LinkedIn, betfair etj. (AP)

4.1. Ruajtja e të dhënave

MongoDB

MongoDB nuk përdor një skemë strikte për të ruajtur të dhënat, por i ruan ato si dokumenta në format BSON. Dokumentat nuk kanë një strukturë të paracaktuar dhe mund të kenë numër të ndryshme fushash. Dokumentat që kanë një strukturë të ngjashme ruhen në njësi qe quhen "Collections". Po të bëjmë një analogji me bazat e të dhënave relacionale do të kishim Collection = Tabelë, dokument = rresht, fusha=kolona.

CouchDB

- I ruan të dhënat si dokumenta në format JSON

4.2. Gjuha e query, pavarësia nga platforma, suporti

MongoDB

- Është shkruar në C++
- Queriet realizohen nëpërmjet metodave dhe funksioneve brenda API të gjuhëve specifike të programimit.
- Mund të implementohet në Linux, OS X, Solaris, dhe Windows.
- Gjuhët e programimit që suportohen nga MongoDB: Actionscript, C, C#, C++, Clojure, ColdFusion, D, Dart, Delphi, Erlang, Go, Groovy, Haskell, Java, JavaScript, Lisp, Lua, MatLab, Perl, PHP, PowerShell, Prolog, Python, R, Ruby, Scala, and Smalltalk

CouchDB

- Është shkruar në Ergland
- Përdor HTTP REST API për të ndërvepruar me bazën e të dhënave

- Mund të implementohet në Android, BSD, iOS, Linux, OS X, Solaris, dhe Windows

- Gjuhët e programimit që suportohen nga CouchDB: C, C#, ColdFusion, Erlang, Haskell, Java, JavaScript, Lisp, Lua, Objective-C, OCaml, Perl, PHP, PL/SQL, Python, Ruby, and Smalltalk.

4.3. Shkallëzimi i bazave të të dhënave nosql

Nga vetë përkufizimi i Big Data dy nga sfidat kryesore të saj janë:

- Rritja e volumit të të dhënave.
- Rritja e ngarkesës si pasojë e rritjes së shpejtësisë me të cilën aksesohen të dhënat.

Për ti përballuar këto të dhëna me volum të madh të cilat aksesohen me shpejtësi të madhe egzistojnë dy zgjidhje të cilat njihen si përshkallëzin vertikal dhe horizontal.

- Shkallëzimi Vertikal- Scale up: Shtojmë fuqinë, burimet fizike, kujtesë, CPU të serverit egzistues të bazës së të dhënave për të rritur performancën. Ky llojë shkallëzimi has kufizimet e mëposhtëme:

– Nuk mund të rritet pafundësisht fuqia procesuese

– Kosto e lartë

– Më pak tolerant ndaj dëmtimeve

- Shkallëzimi Horizontal- Scale Out: Shtimi i disa makinave: rritja e kapacitetit duke lidhur disa entitete software/hardware në mënyrë të tillë që të funksionojnë si një njësi e vetme llogjike. Ky lloj shkallëzimi ka disa avantazhe kundrejt atij vertikal.

– Me pak i kushtueshëm

– Lehtësi për upgrade

MongoDB dhe CouchDB janë sisteme të pështkallëzueshme. Përshkallëzimi realizohet nëpërmjet replikimit dhe particionimit.

- MongoDB favorizon konsistencën e të dhënave
- CouchDB favorizon disponushmërinë e të dhënave

4.4. Replikimi

MongoDB

- Ofron replikimin single-master
- Replikimi asinkron
- Shkrimi- Veprimet e shkrimit ndodhin vetëm te nyja primare
- Përzgjedhje automatike të nyjes primare
- Leximi

– Konsistence strikte: leximi bëhet nga nyja primare default-arrihet CP

– Konsistencë përfundimtare: mund të lexohet edhe nga nyjet sekondare, përcaktohet nga klienti- AP

- Replikimi përdoret të ofruar tepëri e cila na mundëson

– Rekuperim nga dëmtimet HW

– Rekuperim nga ndërprerjet e shërbimeve

CouchDB

- Ofron replikimin master-slave dhe master-master
- Përdor modelin e konsistencës përfundimtare duke favorizuar më shumë disponueshmërinë e të dhënave sesa konsistencën strikte.
- Nyjet sinkronizohem me njëra-tjetrën duke përdorur replikimin master-master
- Sistemi kopjon në mënyrë inkrementale ndryshimet e dokumentave duke bërë që në fund ata të jenë të sinkronizuar.
- Sistemi dedekton dhe zgjidh automatikisht konfliktet gjatë replikimit
- Leximi dhe shkrimi: mund të ndodhin te secila prej nyjeve pa pritur që nyjet e tjera të bien në një marrëveshje.

– Sakrifikohet konsistenca

– Favorizohet disponueshmëria

Cilin sistem duhet të përdorim?

Zgjedhja e një sistemi NoSQL varet nga prioritet e projektit në të cilin ai do të përdoret. Në qoftë se projekti përmban p.sh. të dhëna financiare dhe dëshirojmë një pamje konsistente për të gjithë klientët atëherë MongoDb do të ishte një mundësi e mirë. Në qoftë se mund të tolerohet që disa klientë po shikojnë të

dhëna që nuk janë më të përditësuarat por ju intereson disponueshmëria atëherë CocuchDB do të ishte një zgjedhje e mirë.

4.5. Particionimi

MongoDB & CouchDB për particionimin e të dhënave përdorin atë që njihet si Ranged Sharding: ruajtja e të dhënave në disa makina.

Ndarja e të dhënave bëhet në intervale të vazhdueshme në bazë të vlerës së një çelësi. Çdo shard është një databazë e pavarur. Nëpërmjet particionimit:

- Reduktohet numri i veprimeve në një nyje
- Reduktohet sasia e të dhënave në një nyje

Konkluzione

Është pothuajse e pamundur të përcaktohet në qoftë se një system NoSQL është më i mire sesa një tjetër, kjo gjë varet nga nevojat e aplikacionit jonë. Ne mund të përcaktojmë se cilat kritere duhen përdorur për të diferencuar bazat e të dhënave Nosql:

- Modelimi i të dhënave
- Kompromisi CAP
- Strategjia e replikimit
- Particionimi i të dhënave
- Kompleksiteti dhe gjuha e query

Përdorimi i një databaze Nosql dokument, është një zgjidhje e mirë kur nuk e njohim ekzaktësisht strukturën e të dhënave dhe se si ajo mund të ndryshojë.

Nuk mund të gjejmë një databazë NoQSL të përshtatshëm për çdo rast përdorimi, Aplikacioneve të ndryshme i nevojiten databaza të ndryshme nosql.

Bazat e të dhënave NoSQL implementojnë konsistencën perfundimtare: në një kohë të caktuar replikat mund të mos përmbajnë të njëjtat të dhëna, por përfundimisht ato do të jenë konsistente.

NoSQL kanë kuptim të ndryshëm për njerëz të ndryshëm, ato duhet të krahasohen në varësi të karakteristikave që na nevojiten.

Literatura

Sumit Pal (2016): SQL on Big Data: Technology, Architecture, and Innovation. Apress: 17-35

Eric A Brewer: Towards robust distributed systems (keynote). In 19-th ACM Symposium on Principles of Distributed Computing (PODC), July 2000

Eric A Brewer: CAP twelve years later: How the "rules" have changed. IEEE Computer Magazine: 23-29

Sourav Mazumder, Robin Singh Bhadoria, Ganesh Chandra Deka (2017): Distributed Computing in Big Data Analytics: Concepts, Technologies and Applications. Springer: 20-25

Ganesh Chandra Deka: NoSQL: Database for Storage and Retrieval of Data in Cloud. CRC Press: 100-115

Guy Harrison (2017): Next Generation Databases: Nosql and BigData. Apress: 4-103

CouchDB: http://couchdb.apache.org/

mongoDB: https://www.mongodb.com/