

UNIVERZITET SINGIDUNUM
BEOGRAD
DEPARTMAN ZA POSLEDIPLOMSKE STUDIJE

DOKTORSKA DISERTACIJA

**Upoređivanje performansi analitičkih poslovnih
informatičkih sistema**

MENTOR:

Prof. dr Petar Spalević

STUDENT: Slaviša Ilić

BROJ INDEKSA: 465018/2018

Beograd, 2024. god.

Mentor:

Prof. dr Petar Spalević, mentor, redovni profesor

Univerziteta Singidunum

Članovi komisije:

1. Prof. dr Petar Spalević, mentor, redovni profesor Univerziteta Singidunum
2. Prof. dr Mladen Veinović, redovni profesor Univerziteta Singidunum
3. Prof. dr Stefan Panić, redovni profesor Prirodno-matematičkog fakulteta Univerziteta u Prištini sa privremenim sedištem u Kosovskoj Mitrovici

Datum odbrane disertacije:

Zahvalnica

Zahvaljujem se mentoru prof. dr Petru Spaleviću kao i komentorima prof. dr Mladenu Veinoviću i prof. dr Stefanu Paniću na dragocenim savetima tokom izrade ove doktorske disertacije.

Takođe, zahvaljujem se prof. dr Siniši Iliću na ekspertskim savetima iz oblasti baza podataka i kontinuiranoj podršci.

Posvećeno porodici

Rezime

U današnjoj eri velike količine podataka (*Big Data*), brzina donošenja poslovnih odluka koje su bazirane na podacima je presudna za opstanak i razvoj preduzeća. Nove tehnologije su doprinele nizu promena, počevši od promene načina istraživanja tržišta, do novog pristupa u dizajnu novih proizvoda i njihovom plasiranju na tržištu. Razvoj Interneta, socijalne mreže i novi načini komunikacije su doveli do generisanja enormne količine podataka koje mogu da pomognu preduzećima da iz njih izvuku korisne informacije i donesu prave odluke za svoju budućnost. Jedan od najvećih izazova današnjih preduzeća je kako da pametno investiraju u odgovarajuća IT rešenja koja će im omogućiti generisanje pravih poslovnih informacija iz velikih količina podataka. Podaci se mogu nalaziti na brojnim izvorima, kako u okviru informacionog sistema preduzeća, tako i na Internetu. Rešenja koja se danas nude na tržištu su različita. Jedni od najbitnijih parametara za donošenje odluka preduzeća o izboru odgovarajućeg rešenja su karakteristike, performanse i cena.

Moderan informacioni sistem za poslovno izveštavanje se sastoji od više rešenja koja rade zajedno. Merenje i upoređivanje performansi različitih rešenja koja uključuju sisteme za upravljanje bazama podataka, *Data Warehouse* rešenja, alate za manipulaciju podacima (ETL) i alate za prikaz informacija dobijenih iz podataka (BI) su predmet istraživanja i zaključaka dobijenih na osnovu sprovedenih eksperimenata. Svaki zaključak uključuje i odgovarajuće preporuke za poslovna preduzeća u smislu smernica za izbor odgovarajućeg rešenja.

Ključne reči: Performanse baza podataka, performanse ETL alata, performanse BI alata, sistemi za upravljanje bazama podataka, ETL, *Business Intelligence*.

Naučna oblast: Baze podataka

Sadržaj

REZIME.....	4
SADRŽAJ.....	5
LISTA SKRAĆENICA.....	7
1. UVOD	9
1.1. IDENTIFIKACIJA PROBLEMA I PREDLOG REŠENJA	10
1.2. DOPRINOS DISERTACIJE.....	12
2. STRUKTURA DISERTACIJE	13
3. BAZE PODATAKA	14
3.1. POREĐENJE PERFORMANSI RELACIONIH SISTEMA ZA UPRAVLJANJE BAZAMA PODATAKA	17
3.2. POREĐENJE PERFORMANSI NOSQL I RELACIONIH SISTEMA ZA UPRAVLJANJE BAZAMA PODATAKA	27
3.3. POREĐENJE PERFORMANSI RAZLIČITIH NOSQL SISTEMA ZA UPRAVLJANJE BAZAMA PODATAKA.....	36
3.4. PREGLED, MOGUĆNOSTI I PERFORMANSE ETL ALATA	41
3.5. PREGLED, PRIMENA I KARAKTERISTIKE BI ALATA	47
4. POREĐENJE BRZINA IZVRŠENJA UPITA NAD VELIKIM KOLIČINAMA PODATAKA KORISTEĆI	
RAZLIČITE SISTEME ZA UPRAVLJANJE BAZAMA PODATAKA	52
4.1. PLATFORME BAZA PODATAKA KORIŠĆENE U EKSPERIMENTIMA	54
4.1.1. HP Vertica baza podataka	56
4.1.2. Oracle 11g R2 sa Data Warehouse šablonom i 12c sa In-Memory Column-Store.....	57
4.1.3. MS SQL sa ColumnStore Indeksom	58
4.1.4. MariaDB ColumnStore.....	59
4.1.5. MySQL sa InnoDB i InfoBright engine.....	59
4.2. PRIPREMA TEST OKRUŽENJA	60
4.3. REZULTATI ISTRAŽIVANJA	64
4.4. ZAKLJUČAK	69
5. POREĐENJE BRZINA IZVRŠENJA ETL UPITA PRI KORIŠĆENJU SQL KODA, SSIS UGRAĐENOG	
ETL ALATA I PENTAHO BESPLATNOG ETL ALATA	71
5.1. OKRUŽENJE ZA IZVOĐENJE EKSPERIMENTATA	73
5.1.1. ETL rešenje 1 – korišćenje SQL upita.....	76
5.1.2. ETL Rešenje 2 - korišćenje SSIS funkcionalnih komponenti	77
5.1.3. ETL Rešenje 3 - korišćenje Pentaho ETL alata	80

5.2.	REZULTATI EKSPERIMENTATA.....	85
5.3.	ZAKLJUČAK	87
6.	PERFORMANSE BI ALATA	88
6.1.	OKRUŽENJE ZA IZVOĐENJE EKSPERIMENTATA	90
6.1.1.	<i>BI rešenje 1 – Power BI Desktop</i>	<i>93</i>
6.1.2.	<i>BI Rešenje 2 – Pentaho Analytics.....</i>	<i>97</i>
6.2.	REZULTATI EKSPERIMENTATA.....	101
6.3.	ZAKLJUČAK	103
7.	ZAKLJUČAK	104
8.	LITERATURA.....	107
	SPISAK TABELA	120
	SPISAK SLIKA	121
	SPISAK GRAFIKONA	122

Lista skraćenica

DBMS (eng. Database Management System)	Sistem za upravljanje bazama podataka (srb. SUBP)
OLTP - (eng. Online Transaction Processing)	Transakciono procesiranje u realnom vremenu
OLAP - (eng. Online Analytical Processing)	Analitičko procesiranje u realnom vremenu
ETL - (eng. Extract – Transform – Load)	Ekstrakcija – transformacija - učitavanje
DW - (eng. Data Warehouse)	Skladište podataka
Data Mart	Jednostavna forma skladišta podataka (DW) koja se fokusira samo na jednu oblast, kao što je prodaja ili marketing.
Data Mining	Proces otkrivanja šablona u velikim skupovima podataka, korišćenjem metoda mašinskog učenja, statistike i sistema baza podataka.
BI - (eng. Business Intelligence)	Poslovna inteligencija/poslovno informisanje
NoSQL Database	Baza podataka koja nije zasnovana na SQL jeziku
CPU - (eng. Central Processing Unit)	Centralna procesorska jedinica
GPU - (eng. Graphics Processing Unit).	Grafička procesorska jedinica
RAM - (eng. Random Access Memory)	Memorija sa slučajnim pristupom
DRTDBS - (eng. Distributed Real-time Database Systems)	Distribuirani sistemi baza podataka koji funkcionišu u realnom vremenu
DML - (eng. Data Manipulation Language)	Programski jezik za manipulaciju podataka
Hadoop / Apache Hadoop	Hadup predstavlja softversku biblioteku koja omogućava distribuirano procesiranje velikih setova podataka koristeći klastere računara
XML - (eng. Extensible Markup Language)	Programski jezik koji definiše pravila za opis podataka
JSON	Otvoreni standard za definisanje formata fajlova i podataka koji koristi čitljiv tekst
UML - (eng. <i>Unified Modeling Language</i>)	Jezik za vizualno modeliranje
CRUD - (<i>CREATE, READ/RETRIEVE, UPDATE, DELETE</i>)	Četiri osnovne operacije u bazama podataka: kreiraj, čitaj/pretraži, ažuriraj, briši.

CSV - (eng. comma-separated values)	Tekstualni fajl format koji koristi zareze za odvajanje vrednosti
OS - (eng. Operating System)	Operativni sistem
OLE DB	Aplikativni programski interfejs razvijen od strane kompanije Majkrosoft, koji omogućava pristup podacima na uniforman način.
JDBC (eng. Java DataBase Connectivity)	Predstavlja aplikativni programski interfejs koji omogućava pristup bazi podataka u java programskom okruženju.
SSD (eng. Solid State Drive)	Poluprovodnički disk za smeštanje informacija koji ne koristi pokretne delove.
KPI – Key Performance Indicators	Ključni indikatori performansi
AI - (eng. Artificial Intelligence)	Veštačka inteligencija

1. Uvod

U savremenom poslovanju, brza evolucija tehnologije i sveprisutna digitalizacija transformišu način na koji organizacije posluju i donose ključne odluke. Korišćenje informacija iz podataka postaje osnovna pretpostavka uspešnog poslovanja, otvarajući vrata novim mogućnostima, ali istovremeno postavljajući izazove pred organizacije koje žele da ostanu konkurentne. Ovaj rad je rezultat istraživanja performansi sistema za upravljanje bazama podataka i alata za transformaciju i vizualizaciju podataka u kontekstu korišćenja informacija u donošenju odluka.

Digitalna transformacija donosi sa sobom ogromne količine podataka, ali ključno pitanje je kako organizacije mogu iskoristiti tu enormnu količinu informacija radi ostvarivanja konkurentske prednosti. U eri *Big Data*, analitika podataka postaje imperativ. Poslovni lideri shvataju da nije dovoljno samo prikupiti podatke, već i razviti sposobnost analize i izvlačenja značajnih saznanja iz tih podataka. Kroz pravilnu interpretaciju podataka organizacije mogu identifikovati trendove, prepoznati prilike za inovacije i anticipirati izazove, što im omogućava donošenje pravih poslovnih odluka.

Međutim, uprkos ogromnom potencijalu postoje i ozbiljni izazovi u korišćenju informacija i baza podataka. Pitanja privatnosti podataka, etičkih normi i sigurnosnih pretnji postaju sve relevantnija. Organizacije moraju pažljivo balansirati između upotrebe podataka i zaštite privatnosti svojih klijenata, čime se stvara kompleksan okvir u kom posluju. Osim toga, potrebno je osigurati integritet podataka, kako bi se izbegle pogrešne interpretacije koje mogu dovesti do donošenja pogrešnih odluka.

Današnje poslovanje je neodvojivo povezano sa sposobnošću efikasnog korišćenja informacija i baza podataka. Organizacije koje uspešno savladaju izazove digitalne transformacije i postanu agilne u korišćenju podataka, imaju priliku da oblikuju budućnost svoje industrije. Međutim, da bi se postigao ovaj cilj, potrebno je ne samo investirati u tehnologiju, već i razviti jasne politike, etičke smernice i bezbednosne standarde. Na taj način bi se očuvao integritet i poverenje u korišćenje informacija i baza podataka.

Donosiocima poslovnih odluka nije samo važno da iz podataka dobiju odgovarajuće poslovne pokazatelje, već i koliko se brzo može doći do njih, i po kojoj ceni. Upravo su ova pitanja činila polaznu tačku za istraživanja različitih tehnoloških rešenja koja su predstavljena u ovoj disertaciji.

1.1. Identifikacija problema i predlog rešenja

Digitalna transformacija je prisutna u svim granama industrije sa ciljem unapređenja podrške poslovnim procesima i pružanja podrške donošenju odluka. Od pojave personalnih računara nastala su i softverska rešenja za evidentiranje različitih događaja. Takvi sistemi su na početku bili orijentisani na čuvanje informacija o pojedinačnim događajima. Za kreiranje i generisanje agregacionih izveštaja sa prikazom trendova je u to vreme bilo potrebno napredno informatičko znanje, a sam proces se merio satima a ponekada i danima.

Većina poslovnih organizacija koja bazira svoje odluke na podacima, poseduje dva informaciona sistema za upravljanje podacima: *Online Transaction Processing* (OLTP) i *Online Analytical Processing* (OLAP). OLTP sistem je namenjen evidentiranju poslovnih događaja. On omogućava velikom broju konkurentnih korisnika sistema da pristupe, evidentiraju nove događaje, vrše pretragu i ažuriraju postojeće evidencije. Ovaj sistem je dizajniran za brz unos i ažuriranje podataka, kao i za izveštavanje o pojedinačnim događajima. Ukoliko bi bilo potrebno kreirati složeniji agregacioni upit nad većim brojem tabela i velikom količinom podataka radi generisanja izveštaja u ovom sistemu, takva aktivnost bi proizvela značajan pad performansi sistema koji bi svakako osetili ostali korisnici sistema. Za razliku od OLTP sistema koji je optimizovan za unos podataka, OLAP sistem je dizajniran za analizu i pripremu podataka za generisanje kompleksnih agregacionih izveštaja. Jedan od prvih koraka u projektima naprednog izveštavanja je identifikacija i transformacija podataka iz OLTP sistema, kao i njihovo učitavanje u OLAP sistem koristeći odgovarajući *ETL* (*Extract – Transform – Load*) alat. Ovaj proces se izvodi periodično kako bi OLAP sistem koristio najsvežije podatke. OLAP sistem je unapred pripremljen da skladišti podatke koristeći odgovarajuću arhitekturu koja omogućava izvršavanje potrebnih analiza po različitim parametrima (dimenzijama), velikom brzinom. Ovi sistemi mogu da obezbede različite analize poslovanja kao što su na primer: vrednosti prodaje, nabavke po regionima, grupisane po poslovnicama, po godišnjim kvartalima, po danima u nedelji i slično. Takođe, oni omogućavaju i proučavanje skrivenih zakonitosti u poslovanju. Optimizovani su za brzo pretraživanje i analizu podataka. Podaci se u OLAP sistemima čuvaju u tzv. skladištima podataka (*Data Warehouse - DW*), a korisnicima (najčešće rukovodiocima firme) se analiza podataka i rezultati prikazuju preko softvera/alata poslovne inteligencije (*Business Intelligence - BI*). Ovi alati nude ne samo tabelarni, već i grafički prikaz rezultata sa dinamičkim opcijama filtriranja. Savremeni BI alati su prilično jednostavni i ne zahtevaju poznavanje programskih jezika.

Poslovnim organizacijama su informacije iz OLAP sistema podataka potrebne čim je pre moguće, kako bi ostvarile odgovarajuće prednosti u odnosu na konkurenciju. Iz toga razloga performanse sistema imaju jednu od ključnih uloga u izboru odgovarajućeg rešenja. Nije

svejedno da li će vreme potrebno za dobijanje rezultata jednog upita za analizu trajati nekoliko sati, sat vremena, nekoliko minuta ili nekoliko sekundi.

Predmet i rezultati istraživanja koji su predstavljeni u ovoj disertaciji se odnose na analizu performansi različitih dostupnih rešenja koji su sastavni deo projekata implementacije naprednih rešenja za poslovno izveštavanje i to:

- analiza performansi *Open Source/besplatnih* i komercijalnih *Data Warehouse* sistema koji se koriste za OLAP informacione sisteme
- analiza performansi *Open Source/besplatnih* i komercijalnih ETL alata kojim se vrši izdvajanje podataka iz OLTP baze, transformacija podataka i punjenje OLAP baza,
- analiza performansi *Open Source/besplatnih* i komercijalnih Business Intelligence sistema za povlačenje podataka iz *Data Warehouse* sistema i za prikazivanje rezultata analize.

Na osnovu izvršenih istraživanja, za svaki od navedenih komponenti su izvedeni odgovarajući zaključci i date preporuke u vezi izbora odgovarajućih rešenja, imajući u vidu različite aspekte i zahteve današnjeg poslovanja.

1.2. Doprinos disertacije

U ovom radu su predstavljeni sledeći naučni doprinosi:

- U disertaciji je detaljno naveden pregled dosadašnjih istraživanja. Analizirane su 122 reference u kojima su navedena istraživanja primene, performansi, upoređivanja performansi različitih informacionih sistema na različitim platformama, bilo da su oni u oblaku (*Cloud Based*), na virtualnim mašinama, na određenom operativnom sistemu, na kombinaciji operativnih sistema, na komercijalnim ili *Open Source/besplatnim* softverima i to:
 - ✓ poređenje performansi relacionih sistema za upravljanje bazama podataka,
 - ✓ poređenje performansi NoSQL i relacionih sistema za upravljanje bazama podataka,
 - ✓ poređenje performansi različitih NoSQL sistema za upravljanje bazama podataka,
 - ✓ pregled, mogućnosti i performanse ETL alata,
 - ✓ pregled, primena i karakteristike BI alata.

Na osnovu analize postojeće literature dobijena je polazna osnova za dalje istraživanje.

- Izvršena je analiza karakteristika često korišćenih *Open Source/besplatnih* i komercijalnih *Data Warehouse* sistema, sa posebnim akcentom na konfigurisanje određenih podešavanja sistema (memorija, *row-oriented*, *column-oriented*, *in-memory*) i pokazani su rezultati merenja performansi tih sistema uz različita podešavanja.
- Navedeni su često korišćeni *Open Source/besplatni* i komercijalni ETL alati, pokazani su dobijeni rezultati merenja performansi tih alata i izvršena je analiza dobijenih rezultata.
- Izvršena je analiza karakteristika *Open Source/besplatnih* i komercijalnih softvera za poslovnu inteligenciju (*Business Intelligence*), merenje njihovih performansi i predstavljena je analiza dobijenih rezultata.

Na osnovu svih dobijenih rezultata date su preporuke za poslovne firme, koje alate treba izabrati u odnosu na karakteristike kao što su: performanse, mogućnost integracije sa drugim sistemima itd. Najznačajnije preporuke su date u zaključcima pojedinih sekcija disertacije kao i u krajnjem zaključku ove disertacije.

2. Struktura disertacije

U uvodnom delu disertacije u poglavlju 1 je predstavljen problem koji je predmet istraživanja, kao i način metodološkog pristupa. U disertaciji je izložen osnovni pogled na sisteme za upravljanje bazama podataka, od njihovog nastanka do danas. Pregled dosadašnjih istraživanja se zasniva na prikupljenim podacima o performansama i upoređivanju performansi različitih informacionih sistema na različitim platformama.

Dosadašnja istraživanja su predstavljena u poglavlju 3 i podeljena su u pet sekcija prema oblastima istraživanja:

- Poređenje performansi relacionih sistema za upravljanje bazama podataka,
- Poređenje performansi NoSQL i relacionih sistema za upravljanje bazama podataka,
- Poređenje performansi različitih NoSQL sistema za upravljanje bazama podataka,
- Pregled, mogućnosti i performanse ETL alata,
- Pregled, primena i performanse BI alata.

Navedena istraživanja su poslužila za definisanje polaznih osnova za dalja istraživanja predstavljena u ovoj disertaciji.

Poređenje brzina izvršenja upita nad velikim količinama podataka koristeći različite sisteme za upravljanje bazama podataka je predstavljeno u poglavlju 4. Izvršena je analiza i merenje karakteristika često korišćenih komercijalnih i *Open Source/besplatnih Data Warehouse* sistema sa posebnim akcentom na konfigurisanje i rezultate određenih podešavanja, kao što su memorija, *row-oriented*, *column-oriented*, *in-memory* i slično.

Istraživanje poređenja performansi najčešće korišćenih komercijalnih i *Open Source/besplatnih* ETL alata kao i analiza dobijenih rezultata je predstavljena u poglavlju 5.

U poglavlju 6 je predstavljena analiza performansi komercijalnog BI alata Microsoft PowerBI Desktop i besplatnog BI alata Pentaho Analytics, na osnovu izvršenih eksperimenata.

U poglavlju 7 je dat sveobuhvatni zaključak koji obuhvata navedena istraživanja iz oblasti *Data Warehouse* sistema, ETL i BI alata, kao i pravci daljih istraživanja. Zaključak obuhvata i preporuke za izbor odgovarajućih rešenja namenjenih poslovnim preduzećima.

Korišćena literatura je navedena u poglavlju 8.

3. Baze podataka

Tipovi baza podataka se često nazivaju i modelima baza podataka i predstavljaju odgovarajuće strukture koje se koriste za organizovanje podataka u okviru Sistema za upravljanje bazama podataka (SUBP) ili na engleskom *Database Management System (DBMS)*. Postoje različiti tipovi baza podataka. Neke baze podataka možemo da posmatramo iz istorijske perspektive jer danas više nisu u upotrebi, dok su ostale koje su u upotrebi, spremne da reše izazove novijih zahteva i očekivanja od strane krajnjih korisnika i različitih industrija.

Stariji tipovi baza podataka kao što su *Flat File* baze (čuvaju podatke u okviru jednog fajla u formatu čistog teksta), hijerarhijske baze podataka (koriste relacije roditelj-dete da bi povezale podatke u strukturu stabla), ili pak mrežne baze podataka (omogućavaju kreiranje kompleksnijih relacija podataka), predstavljaju neku vrstu evolucije ka razvoju sistema za upravljanje podacima, koje imamo danas u savremenoj upotrebi.

Prekretnicu u razvoju baza podataka su predstavljale relacione baze podataka koje koriste tabele sa odgovarajućim vezama za smeštanje i upravljanje dobro struktuiranim podacima. Svoju primenu su našle u skoro svim industrijama i mogu biti veoma robusne. Neki od predstavnika ovih SUBP su: MySQL, Oracle, IBM DB2, Microsoft SQL Server, MariaDB, PostgreSQL i druge.

Za potrebe rešavanja izazova za smeštanje i manipulaciju ogromnih količina nestruktuiranih i polustruktuiranih podataka na internetu, početkom 2000-ih godina su se pojavile takozvane NoSQL baze podataka, često nazivane "*non-SQL*" ili "*not only SQL*". One predstavljaju alternativu za upravljanje podacima koje je teško smestiti u tradicionalni relacioni sistem. Postoji nekoliko tipova NoSQL baza podataka koje imaju svoju primenu u rešavanju savremenih izazova, od kojih su najčešće korišćeni:

- *Key-value* baze podataka, koje koriste jednostavne skladišne ćelije sa označenim ključem, kojima se može pristupiti od strane distribuiranih sistema ili serverskih klastera. Najčešća primena ovih baza podataka je u scenarijima kada je potrebna brza manipulacija nad podacima, kao što su konfiguracioni podaci, informacije o statusnim stanjima i slično. Predstavnici ovih baza podataka su: Azure Table Storage, MEMBASE, Riak, Redis, Memcached, Etc.
- *Document* baze podataka koje se često nazivaju i JSON (JavaScript Object Notation) baze, smeštaju podatke u fleksibilne i samo-opisujuće strukture u formi dokumenata. Podaci se smeštaju u kolekcije dokumenata. Svaki dokument predstavlja zaseban entitet i sadrži podatke i asocirane metapodatke.

Predstavnici ovih baza podataka su: CouchDB, MongoDB, Document-based MySQL, Terrastore, RethinkDB.

- *Graph* baze podataka su namenjene skladištenju ogromnih količina podataka na internetu. Podaci se smeštaju u nodove koji predstavljaju entitete ili objekte, i ivice (edges) koje predstavljaju relacije između nodova. Dizajnirane su za skladištenje podataka sa kompleksnim relacijama i koriste se za smeštanje podataka socijalnih mreža, sistema za detekciju prevara i drugo. Predstavnici su: Neo4J, Infinite Graph, Sones, InfoGrid, JanusGraph, Dgraph.
- *Column* baze podataka ili *Wide Column-store / Column Families* su dizajnirane da reše izazove koji su nastali u praznom prostoru funkcionalnosti između relacionih i *document* baza podataka. U relacionim bazama podataka, šema definiše kolone u tabeli, tip podataka svake kolone, kao i druge kriterijume. Svi redovi u tabeli moraju da odgovaraju definisanoj šemi. *Column* baze podataka, koriste strukturu familija kolona (*column families*) čiji redovi koriste svoje zasebne formate. Svaki red se sastoji od jedinstvenog identifikatora reda i skupa imena kolona i njihovih vrednosti. Koristeći ovakav dizajn, svaki red u familiji kolona ima svoju šemu. Šema se može lako modifikovati ukoliko se za tim ukaže potreba. Predstavnici ovih baza podataka su: Hadoop / HBase, Cassandra, Hypertable, Cloudata, Amazon SimpleDB, SciDB.

Takođe, treba napomenuti i neke dodatne tipove baza podataka koje zbog svojih specifičnih karakteristika mogu predstavljati odličan izbor:

- *Column-oriented* baze su vrlo slične relacionim bazama podataka, samo što se kod njih podaci smeštaju na disk po kolonama, umesto po redovima. To u praksi znači da su svi podaci jedne kolone u tabeli smešteni zajedno, čime se omogućava brza agregacija većih skupova podataka. Ovaj tip baze podataka predstavlja odlično rešenje za analitičke potrebe. Radi izbegavanja konfuzije, treba potpuno razdvojiti *Column-oriented* baze podataka predstavljene u ovom pasusu i prethodno navedene NoSQL *Column* baze.
- *Objektno orijentisane* baze podataka, kod kojih su podaci organizovani u objekte. Objekti sadrže kako same podatke, tako i metode za izvršavanje operacija nad tim podacima. Dizajnirane su da podrže veoma kompleksne strukture podataka i njihova najšira upotreba je od strane aplikacija koje zahtevaju pristup velikoj količini struktuiranih ili polustruktuiranih podataka.

U današnjem poslovanju jedna od bitnih karakteristika za izbor odgovarajuće platforme baza podataka su njene performanse. Imajući u vidu prisutnost različitih tipova baza podataka koje mogu biti korišćene kao najbolji izbor za specifične izazove i zahteve, brojni istraživači su pokušavali da daju odgovor u vidu odgovarajućih smernica na osnovu izvršenih eksperimenata.

U narednim poglavljima je izložen pregled dosadašnjih istraživanja vezanih za ispitivanje performansi i upoređivanje performansi različitih informacionih sistema (instaliranih na različitim platformama), bilo da su oni u oblaku (*Cloud-Based*), na virtualnim mašinama, na određenim operativnim sistemima, na kombinaciji operativnih sistema, na komercijalnim ili *Open Source/besplatnim* softverima.

3.1. Poređenje performansi relacionih sistema za upravljanje bazama podataka

Često korišćen termin u eksperimentima poređenja performansi različitih sistema je “Benčmark“. On u svom opštem značenju predstavlja proces upoređivanja performansi, procesa ili praksi organizacije sa onima koji se smatraju najboljima u industriji ili sličnim organizacijama. Benčmark baza podataka predstavlja standardni skup izvršnih instrukcija koje se koriste za merenje i upoređivanje performansi različitih sistema baza podataka, izvođenjem kontrolisanih eksperimenata.

Autori su u radu [1] predstavili sintetički prototip metod koji pored standardnih benčmark eksperimenata koristi i dodatni set od više generalizovanih korisničkih eksperimenata. Pod standardnim benčmark testovima se podrazumevaju: TP1 (dizajniran za merenje kompletne transakcije), TPC-A (dizajniran za merenje broja transakcija u sekundi koje sistem može da izvrši kada se pokreće sa više terminala), TPC-B (dizajniran za merenje propustnosti, npr. koliko transakcija u sekundi dati sistem može da izvrši), TPC-C (dizajniran za on-line obradu transakcija koja uključuje skup od pet istovremenih transakcija različitih tipova i složenosti, a koje se mogu izvršiti na mreži ili u redu za odloženo izvršenje), TPC-D (dizajniran za podršku u odlučivanju koji modelira nekoliko opšte primenljivih aspekata sistema za podršku odlučivanju, a uključuje upite i održavanje podataka), TPC-H (dizajniran za obradu transakcija baze podataka, specifičan za podršku u odlučivanju), TPC-R (dizajniran za podršku u odlučivanju - sličan TPC-H, koji omogućava dodatne optimizacije na osnovu korišćenja naprednijih upita i istovremenih modifikacija podataka), TPC-VMS (dizajniran za testiranje baza podataka koje rade u režimu virtualizacije), TPC-W (dizajniran za transakcioni veb benčmark, kod koga se radno opterećenje obavlja u kontrolisanom okruženju internet trgovine). Prototip je kreiran sa korisničkim interfejsom radi unosa dodatnih zahteva. Autori zaključuju da test rezultati standardnih benčmarka predstavljaju procene mogućih performansi sistema za određene predefinisane tipove aplikacija i da se performanse sistema baza podataka na aktuelnom domenu baze mogu značajno razlikovati od onih koji su predviđeni standardnim benčmarkom. Rezultati grupa standardnih TPC benčmark testova postaju validni samo ako se testiranje obavlja pod propisanim uslovima, kao što su: količina podataka u bazi mora biti veća od propisane, opterećenje baze mora biti veće od propisanog, broj redova u tabelama nad kojima se vrši testiranje mora biti veći od definisanog broja, itd. Iz ovoga se jasno vidi da se TPC testovi ne mogu izvršiti nad svakim sistemom.

Informacioni sistemi baza podataka su se ranije tradicionalno pokretali (u vreme pisanja ovog rada) na jednom host kompjuteru, na kome se nalazio operativni sistem, aplikacioni server i server baze podataka [2]. Kod ovakve arhitekture, postojanje servera baze podataka zajedno sa aplikacionim serverom na istom računaru može značajno da degradira performanse host

računara. Arhitektura sa bazom podataka na zasebnom hostu (*backend database machine architecture*) je predložena radi premeštanja opterećenja sa zajedničkog hosta na zaseban specijalizovan kompjuterski sistem. Na ovaj način se ukupno opterećenje deli na dva računarska sistema, aplikativni koji generiše zahteve ka sistemu baza podataka i sistem baza podataka koji procesira zahteve nad bazom podataka. Analiza prednosti i mana ove arhitekture uključuje razmatranja parametara kao što su performanse, pouzdanost, deljenje baze, bezbednost i modularnost. Iako je povećanje performansi sistema ključni prioritet, osnovni izazov se ogleda u prihvatljivom odnosu „cena/performance“. Istraživači na Univerzitetu u Viskonsinu su razvili sveobuhvatnu strategiju za testiranje performansi sistema za upravljanje bazama podataka (SUBP) u arhitekturi gde je SUBP instaliran na istom hostu sa aplikativnim serverom i arhitekturi gde je SUBP instaliran na zasebnom računaru. Kako bi se testirale razlike u performansama ove dve arhitekture, kreirana je posebna sintetička baza podataka i izvršeno je *single-user* i *multiple-user* testiranje na sistemima koji uključuju Oracle, Ingres, SQL-DS i IDM-500. Rezultati testiranja su pokazali da arhitektura sa SUBP na zasebnom hostu za kraće vreme prikazuje rezultate upita.

U savremenim informacionim sistemima, baze podataka se mogu koristiti kao OLTP baze, OLAP baze, servisne arhitekture i *e-Science* aplikacije [3]. Zbog velike količine podataka koje se u njima nalaze i velikog broja korisnika koji pristupaju aplikacijama i servisima, često su baze podataka organizovane u distribuirane sisteme. Ti sistemi se nalaze na velikom broju mrežnih čvorova i prilikom izvršavanja upita moraju da razmenjuju podatke preko računarske mreže. Komunikacija koja se dešava između čvorova uključuje procese povezane sa kontrolom konkurentnosti, podešavanjem replikacije, algoritmima distribuiranog ažuriranja i slično. U tim sistemima baze su instalirane na različitim arhitekturama i računarima sa različitim fizičkim performansama, kao što su broj i brzina CPU-ova, količina RAM memorije, veličina i brzina diskova itd. Nad bazama se izvršavaju i transakcije kojima se ažurira određeni broj tabela saglasno algoritmima za kontrolu konkurentnosti. Kako bi se na neki način merile performanse takvih sistema, određena grupa autora je probala da postavi modele i definiše parametre tih modela kako bi se prvo teorijski, a kasnije i simulacijama, na neki način potvrdila valjanost modela.

Jedan od korišćenih modela za analizu performansi softvera je tzv. *Queuing Networks (QN)* [4]. To je model koji rešava zahteve korisnika (*service requests*) koji dolaze u servisne centre (serveri) i formiraju red (*queue*), kako bi bili usluženi. U ovim modelima, kao parametri modela se koriste:

- brzina kojom korisnici dolaze na red (*workload intensity*), npr. svake dve sekunde,
- srednje vreme potražnje usluga od strane korisnika (*average service requirement of a customer*), npr. 1.25 sekundi,
- opterećenje sistema (*utilization*),
- srednje vreme potrebno da se korisnik potpuno usluži (*residence time*),

- broj korisnika koji čekaju na red (*queue length*) i
- propustnost (*throughput*).

Koristeći navedenu metodu, autori u radu [5] povezuju performanse komponenti distribuiranih sistema baza podataka (brzina i broj CPU-ova, brzina diskova, kapacitet računarske mreže, vreme između zaključavanja tabela, broj redova koji se upisuju u tabele i dr.) u gore navedene parametre. Oni su takođe izvršili i analizu velikog broja objavljenih radova u kojima su na različite načine modelovani sistemi sa bazama podataka koristeći QN metodu i pokušali da grupišu te modele u zavisnosti od nivoa detalja kojima je interni dizajn transakcija postavljen u modelu. Tako su svi analizirani modeli iz 30-ak radova podeljeni u: model crne kutije (gde transakcioni dizajn nije predstavljen u modelu), model transakcione obrade (gde je transakcija predstavljena kao odgovor na zahtev za uslugu), model veličine transakcije (gde je model zasnovan na broju objekata na koje transakcija deluje) i model transakcione faze (gde je model zasnovan na broju faza kroz koje transakcija prolazi prilikom izvršavanja zahteva). Za svaki analizirani rad je dato objašnjenje kako je modelovan QN. Zaključak autora je da na performanse sistema baza podataka utiču kompleksnost i međuzavisne funkcionalnosti, kao i referentne lokacije izvršenja transakcija i konkurentnog pristupa memoriji i disku.

Sistem za upravljanje bazama podataka ili skraćeno SUBP ima značajnu ulogu u današnjem poslovanju u mnogim industrijama i projektima. Ovi sistemi se koriste za skladištenje podataka, njihovu manipulaciju i izveštavanje. Postoji niz kompanija koje proizvode SUBP sisteme i oni se na tržištu mogu naći po različitim cenama, od mogućnosti besplatnog korišćenja, u slučaju *Open Source/besplatnih* rešenja, do rešenja u iznosu od više desetina hiljada dolara u slučaju *Proprietary* SUBP rešenja. Potencijalni korisnici ovih rešenja donose odluku o izboru adekvatnog rešenja na osnovu različitih karakteristika koje rešenje nudi (performanse, pouzdanost, bezbednost i dr.). Pored toga bitnu ulogu u izboru rešenja može da ima i njegova cena.

Autori su u radu [6] izvršili merenje performansi pet različitih SUBP dostupnih na tržištu: MS SQL Server 2008, Oracle 11g, IBM DB2, MySQL 5.5 i MS Access 2010. Za potrebe testiranja korišćeni su SQL upiti različite kompleksnosti nad svim navedenim sistemima. *Performance* benčmark je korišćen za merenje vremena izvršenja upita, CPU opterećenja, korišćenje memorije, korišćenje virtuelne memorije, kao i broj izvršenih procesorskih niti (*threads*). Testiranja su izvedena na kompjuterskoj platformi sa Dual-Procesorom, Intel Xeon E5649, 6x2 kora, procesor klok 2.53GHz sa 32GB RAM-a i 2TB sekundarnog skladišnog kapaciteta, koristeći MS Windows Server 2008, 64-bitne verzije. Autori su razvili aplikaciju za testiranje koristeći C#.NET pod NET Framework 4.0. Aplikacija je izvršavala dva zadatka: automatsko popunjavanje tabela baza sa 1,000,000 redova pre izvršenja samog testiranja i izvršavanje SQL upita različitih kompleksnosti. Pored prethodno navedenog, u aplikaciju je ugrađen i tajmer za merenje izvršenja procesa. Za merenje korišćenja memorije i procesorskog vremena je korišćen MS Windows Task Manager (WTM) alat koji dolazi uz MS Windows operativni sistem. Na

svim SUBP platformama autori su koristili isto okruženje sa primerom baze podataka koja se koristi u *retail* poslovanju. Ovo okruženje je uključivalo *Front End* sistem za generisanje faktura, računa i porudžbenica i *Back End* za upravljanje podacima u skladištu. Tabele baze podataka su pre testa popunjene sa po 1,000,000 redova, nasumičnim vrednostima. Za potrebe testa izvršeno je 10 različitih tipova upita i nivoa kompleksnosti. Rezultati testiranja su pokazali da je za date primere testova IBM DB2 pokazao najveću brzinu izvršenja upita, dok je MS Access pokazao najmanje opterećenje procesora u poređenju sa ostalim SUBP sistemima. Prosečno korišćenje memorije je najniže kod MS Access, a najviše kod IBM DB2.

Sistemi baza podataka se mogu klasifikovati kao centralizovani i distribuirani. Dok se procesi kod centralizovanih sistema izvršavaju nad jednom kompjuterskom jedinicom, distribuirani sistemi baza podataka uključuju kompjuterske jedinice na više lokacija povezane komunikacionom mrežom, pri čemu sistem na svakoj lokaciji ima skup svoja pravila, a sistemi na svim lokacijama rade zajedno. Korisnik na bilo kojoj lokaciji može pristupiti i raditi sa podacima koji se fizički nalaze na drugim lokacijama, na način kao da se svi podaci nalaze na lokaciji korisnika. Današnje poslovanje zahteva distribuirano procesiranje podataka iz više razloga, a najviše radi obezbeđenja konkurentnosti na tržištu. Algoritmi koji se primenjuju u sistemima baza podataka za pristup i rad sa podacima moraju da zadovolje vremenske rokove podržanih aplikacija. Autori u radu [7] razmatraju probleme sa performansama koje su značajne za *distributed real-time database systems (DRTDBS)* i istražuju dosadašnja saznanja i istraživanja nad parametrima kao što su sistemi dodele prioriteta procesa, *real-time* izvršni protokoli, kao i optimizacija korišćenja memorije u replikacionim, odnosno ne-replikacionim okruženjima, koji se odnose na procesiranje distribuiranih *real-time* transakcija. Autori na osnovu svog istraživanja zaključuju da polise dodele prioriteta imaju svoj značaj samo kada se kohorti transakcije (skup procesa koji se izvršavaju u ime inicijalne transakcije, na drugim sajtovim) izvršavaju sekvencijalno. Najveći deo *Real Time Commit* protokola pokušava da poboljša performanse, omogućavajući transakcijama da koriste podatke zaključane od strane drugih transakcija. Trenutno, postoji ozbiljan nedostatak protokola koji bi mogli bolje i efikasnije da upravljaju korišćenjem memorije, kreiranjem manjeg broja privremenih objekata.

U toku projekta razvoja rešenja "Repose" za *Resort Management System (RMS)* [8] koji je imao za cilj da skladišti informacije o hotelima, njihovim korisnicima, servisima koji se nude, rezervacijama itd., autori su uvideli priliku za analizu i poređenje performansi dva SUBP i to MySQL i SQL Server. Oni su viđeni kao najuži izbor rešenja za upravljanje i skladištenje podataka. MySQL je jedna od najpopularnijih *Open Source/besplatnih* relacionih SUBP, koja pored osobina kao što su podrška različitim tipovima podataka, podrška zapisima sa fiksnom i promenljivom dužinom, brzo izvršenje procesa itd., nudi i nezavisnost od platforme operativnog sistema. Sa druge strane, SQL Server je SUBP proizvod kompanije Microsoft, dizajniran za rad u *Enterprise* okruženjima sa veoma naprednim karakteristikama kao što su optimizacija memorije u radu sa tabelama i uskladištenim procedurama, obezbeđivanje alata za migraciju podataka sa najkorišćenijih sistema baza podataka, podrška za klaster i replikacione

servise itd. “Repose“ RMS je korišćen kao alat za prikupljanje podataka koristeći dnevnu interakciju sa korisnicima koji su unosili podatke u sistem. Za potrebe testiranja i merenja performansi korišćeni su SQL Server 2008 i MySQL 5.6.17. Podaci su prvo prikupljeni u MySQL a potom migrirani na SQL server koristeći alat Migration Assistant for MySQL. Računar koji je korišćen za uporedno testiranje je 64-bitni Windows 8 sa Intel® Core™ i5-2430M CPU, sa klok frekvencijom 2.4 GHz. i 4GB RAM. Autori su za potrebe testiranja dizajnirali nekoliko različitih DML upita (*SELECT*, *INSERT*, *UPDATE* i *DELETE*) koji su izvršeni na oba *Back-End* SUBP sistema, pri čemu je mereno vreme njihovog izvršenja. Rezultati merenja performansi su pokazali da SQL Server nudi bolje performanse u odnosu na MySQL kada se posmatra parametar “vreme izvršenja“. U svim test slučajevima, izuzev Insert upita, SQL Serveru je bilo potrebno manje vremena za izvršenje. MySQL je takođe pokazao slabije rezultate u smislu skaliranja, skoro dva puta više vremena je bilo potrebno za izvršenje upita u slučajevima povećanja broja redova. Najubedljivija razlika u performansama navedena dva SUBP se vidi kod primene *SELECT* upita nad tabelom sa 3,000 redova, gde je MySQL-u bilo potrebno dvostruko više vremena za izvršenje upita u odnosu na SQL Server. Iako rezultati merenja performansi nedvosmisleno ukazuju da je SQL Server očigledan izbor za *Back-End*, njegova cena predstavlja veliki izazov za mala preduzeća. *Open Source/besplatna* priroda MySQL-a ujedno znači i da će cena implementacije biti minimalna, te iz tog razloga mala preduzeća često biraju ovaj proizvod koji nudi zadovoljavajuće performanse na manjim bazama podataka koje se koriste u web okruženjima.

U *Enterprise* okruženjima za rad sa velikom količinom podataka izbor odgovarajuće relacione SUBP platforme ima veliki značaj. Baze podataka predstavljaju jezgro aplikativnih rešenja u mnogim industrijama. Platforme relacionih baza podataka koje se nude na tržištu mogu biti *Proprietary* za koje je potrebno kupiti licencu (Microsoft, Oracle, IBM) dok ostale mogu biti *Open Source/besplatne*. Izbor relacione baze podataka može biti dobar ukoliko korisnik poseduje ograničenu količinu podataka. Radi postizanja najboljih performansi na izabranoj platformi, dizajneri baze podataka imaju ključnu ulogu, jer će od dizajna baze zavisiti i brzina izvršavanja procesa nad podacima u bazi. Autori su u radu [9] izvršili komparativnu analizu rezultata dobijenih merenjem performansi tri najčešće korišćene serverske baze podataka: SQL Server, Oracle i MySQL. Različiti tipovi upita sa različitim nivoima kompleksnosti su korišćeni za potrebe merenja vremena izvršenja. Svi eksperimenti su izvršeni na Windows operativnom sistemu. Korišćena je metodologija testiranja vremena izvršenja upita na SQL Server 2016, Oracle 12c i MySQL 5.7, pokretanjem različitih tipova upita. Za potrebe eksperimenta autori su limitirali broj redova u tabelama između 300,000 i 400,000. Tačnije 324,500 redova je ubačeno u svaku od navedene tri baze podatka koristeći softver koji je specijalno napravljen za potrebe ovog eksperimenta. Rezultati testiranja nad ovom količinom podataka su pokazali da MySQL pokazuje najbolje performanse kod upita koje nemaju postavljene uslove u upitu. Kod jednostavnih kondicionalnih upita sa *WHERE* klauzulom, MySQL i SQL Server su pokazali dobre i identične performanse za razliku od Oracle, kome je potrebno više vremena. SQL Server

pokazuje najbolje performanse u odnosu na ostala dva rešenja kod naredbi za ažuriranje podataka u jednoj koloni nad svih 324,500 polja u tabeli. Oracle pokazuje najbolje rezultate u odnosu MySQL i SQL Server kod eksperimenta kopiranja kompletne sadržaja predmetne tabele u drugu. Izbor MySQL baze podataka može biti dobar u slučaju malih preduzeća kada količina podataka nije velika, jer u tim slučajevima MySQL pokazuje dobre karakteristike performansi. Kod izbora baze podataka treba imati u vidu i njihovu cenu. MS SQL Server i Oracle baza imaju cene od nekoliko desetina hiljada dolara dok je MySQL besplatan.

Poređenjem performansi relacionih baza podataka, Microsoft SQL Servera i Oracle su se bavili i autori u radu [10], koristeći različite upite nad tabelama od 100,000 i 500,000 redova kao i funkcije kreiranja, ažuriranja i brisanja zapisa iz tabela. Rezultati istraživanja su pokazala da je SQL server 2000 pokazao generalno bolje rezultate od Oracle 9i kod većine upita.

U poslednjih nekoliko dekada relacione baze podataka su dominirale u skoro svim industrijama i predstavljale okosnicu platforme za skladištenje i upravljanje podacima. Relacione baze najčešće upisuju skupove podataka kontinuirano, u redovima na fizičkom disku. Ovakav pristup se obično naziva “*N-ary storage model (NSM)*“. Za razliku od prethodno navedenog, pojedine implementacije relacionih baza koriste takozvani *Column-Store* pristup upisa podataka na disk, odnosno “*decomposed storage model (DSM)*“. Autori su u radu [11] izvršili niz eksperimenata ispitivanja performansi koristeći predstavnike implementacija ova dva pristupa, Oracle i Vertica. Istraživanje su započeli analizom data modela i optimizacijom performansi oba sistema. Potom su razvili benčmark performansi kako bi kvantifikovali performanse sistema sa *row* i *column* orijentacijom podataka i na kraju na osnovu rezultata eksperimenata izveli odgovarajuće zaključke za različite tipove upita i opterećenja. Svaka implementacija relacionih baza podataka definiše kako se podaci fizički smeštaju na disk. Kod Oracle 11g data fajlovi se se smeštaju redno u tzv. tabelarna polja (“*tablespaces*“), pri čemu se svaki data fajl može nalaziti u samo jednom tabelarnom polju, a svako tabelarno polje može da sadrži više data fajlova. Druge značajne strukturne komponente Oracle baze uključuju kontrolne fajlove, arhivirane logove, fajlove za praćenje promena itd. Veličine data blokova kod Oracle su od 2KB do 32KB. Vertica koristi *Column-store* implementaciju upisa podataka. Ona koristi dva tipa skladišta, *Read Optimized Store (ROS)* i *Write Optimized Store (WOS)* za smeštanje podataka i svako skladište ima svoj tip kontejnera. ROS kontejner sadrži skupove kompletnih podataka sortiranih po određenom redosledu, kao par fajlova po koloni. Svaka kolona je smeštena u dva zasebna fajla, jedan sa aktuelnim podacima, a drugi sa pozicionim indeksom. Pozicioni indeks je važan podatak koji značajno utiče na performanse. WOS kontejner sadrži podatke koji se čuvaju isključivo u memoriji, pri čemu njihova orijentacija (u redovima ili kolonama) ima vrlo mali značaj. Autori su eksperimente izveli na serveru sa 4xIntel Xeon procesorima sa po 4 kora, kolokovanim na 2.4 GHz, 128 GB RAM i 4 TB čvrstim diskom. Baze podataka su instalirane na virtualnim mašinama koristeći VMware ESXi softver. Svakoj virtuelnoj mašini je dodeljeno 2 CPU kora, 6 GB RAM-a i 16 GB skladištenog prostora. Za predstavnike baza podataka korišćene su Oracle 11g SE kao Oracle instanca i HP Vertica Community Edition kao Vertica

instanca. Ista šema baza podataka je importovana na obe instance, koristeći produkcionu bazu podataka putničke agencije kao izvor podataka. Iako se za benčmark kod merenja performansi *Column-store* baza obično koristi *TPC-H benchmark* ili *Vertica Star Schema Benchmark* (SSB) koji je izveden iz TPC-H, autori ovog rada su odlučili da koriste svoj benčmark. Razlog za takav izbor se ogleda u značajnoj razlici karakteristika šeme baze podataka koja je korišćena za eksperimente u odnosu na onu koja je propisana gore navedenim benčmarkom. Modeliranje sopstvenog benčmarka je omogućilo autorima ovog istraživanja da uključe upite koje imaju drugačiju namenu u odnosu na SBB i TPC-H benčmarke. Dizajnirani benčmark se sastojao od dva skupa upita, tzv. generalnih upita koji mere performanse/vreme izvršenja jednostavnih operacija (*INSERT*, *UPDATE*, *DELETE* i upita za ne-analitičku namenu) i analitičkih upita koji se koriste za merenje različitih tipova analitičkih zahteva. Rezultati merenja su pokazali da Oracle poseduje značajno bolje performanse u odnosu na Vertica kod operacija manipulacije podacima u bazi i upita koji su ne-analitičkog tipa. Međutim, kada se razmatraju performanse sistema pri izvršenju analitičkih upita, Vertica je pokazala znatno veću brzinu izvršenja upita od Oracle, tj. performanse Vertica su bile od 3 do 32 puta bolje od Oracle u zavisnosti od tipa upita. Imajući u vidu rezultate istraživanja, zaključak je da je sa aspekta performansi Vertica bolji izbor za *Data Warehouse* rešenja, dok je Oracle bolji izbor za tradicionalne transakcione tipove opterećenja.

Ogromno povećanje količine podataka koje je potrebno skladištiti i analizirati donosi potrebu za odgovarajućim alatima, tehnikama i okruženjima za *Big Data* analitiku. Hadoop i Apache Hive su alati koji dizajnirani za rad sa *Big Data* [12]. Apache Hive predstavlja *Open Source* aplikaciju za analiziranje velikih skupova podataka. Možemo reći da je ovaj alat u stvari *Data Warehouse* softver koji omogućava pojednostavljene upite nad podacima, upravlja skupovima podataka u distribuiranim skladištima i koristi Hadoop kao osnovni sloj. Hadoop predstavlja *Open Source* implementaciju *MapReduce* programskog modela koji je napisan u Javi i omogućava distribuirano procesiranje velikih baza podataka koje se nalaze na različitim klasterima. Dizajniran je da bude skalabilan i lako se proširuje od jednog do nekoliko hiljada servera, pri čemu svaki od njih nudi lokalno procesiranje i skladište podataka. Oracle SQL predstavlja tradicionalnu relacionu bazu podataka. Na osnovu informacija iz literature, autori rada su zaključili da je više istraživača pokušalo da izvede *Big Data* analitiku koristeći tradicionalne metode. Ovi pokušaji su rezultirali prilično lošim rezultatima sa aspekta performansi, zbog ograničenja memorije. Iz tog razloga su odlučili istraže i uporede performanse Apache Hive distribuirane baze i popularnog Oracle SUBP. Ukupno četiri parametra su korišćena za određivanje finalnog rezultata i to: broj redova, razmatranje tipova upita, veličina data-setova i poređenje performansi izvršenja upita između Apache Hive i Oracle SQL. Četiri data seta koji su korišćena u eksperimentu su preuzeta su sa *UCI Machine Learning Repository*. Prva dva data seta su sadržala manju količinu podataka, za razliku od druga dva data seta, kako bi se ustanovile karakteristike obe platforme pri obradi različite količine podataka. Autori su definisali 5 različitih upita i testirali ih nad sva četiri data seta na Apache

Hive i Oracle okruženju i to: upit koji vraća jedinstvene zapise, upit sortiranja, upit inverznog sortiranja, upit sa grupisanjem i upit koji vraća maksimalnu vrednost. Zaključak autora na osnovu rezultata eksperimenata je da za procesiranje manje količine podataka Apache Hive zahteva više vremena od Oracle baze, dok je kod velikih data setova rezultat obrnut, u korist Apache Hive.

Istraživanje za izbor odgovarajuće platforme baza podataka na osnovu analize performansi različitih tipova baza podataka u kontekstu podataka zdravstvenog sistema je bio predmet istraživanja u radu [13]. Različiti standardi za elektronske zdravstvene zapise kao što su HL7, OpenEHR i CEN EN 13606 su u istraživanju bili korišćeni za kreiranje objektno orijentisane baze podataka. Do trenutka pisanja ovog rada nijedan od navedenih standarda nije korišćen za kreiranje objektno orijentisane baze podataka. Autori su prvo izvršili analizu trendova u zdravstvenom sistemu Indije i uporedili performanse sistema MySQL baze kao relacione baze i Db4o objektno orijentisane baze za isti uzorak podataka za 100 pacijenata. Rezultati izvedenih eksperimenata su pokazali da su objektno orijentisane baze bolje u poređenju sa relacionim bazama kada se razmatra vreme potrebno za procesiranje podataka, dok sa druge strane objektna baza koristi veću količinu skladišnog prostora u poređenju sa relacionim bazama.

Izborom adekvatne platforme baza podataka za rad sa podacima iz oblasti zdravstvene zaštite se bave i autori u radu [14]. Posle niza godina korišćenja relacionih baza podataka u zdravstvu, za njihovo trajno skladištenje razmotrene su moguće alternative kao što su *triplestores* sistemi (kod kojih se entiteti podataka sastoje od definicija subjekata, predikata i objektnih informacija). Rezultati istraživanja su pokazali da *triplestores* sistemi mogu imati veoma dobre performanse kada se posmatra prosečno vreme skladištenja, odnosno čitanja podataka iz sistema.

Postoje dva načina za mapiranje tabela dvodimenzionalne relacione baze podataka u skladište i to: upis podataka red po red (*row oriented*) ili upis podataka kolona po kolona (*column oriented*). Tradicionalne baze podataka vrše upise po sistemu red po red. Međutim za neke aplikacije koje su analitičkog tipa upis podataka po sistemu red po red nije najbolje rešenje sa aspekta performansi. Autori su u radu [15] izvršili merenje performansi ovih sistema i došli do zaključka da je *column oriented* pristup pokazao bolje rezultate performansi kod *Data Warehouse* sistema, dok je *row oriented* pristup imao bolje performanse kod procesa upisa i ažuriranja baze podataka.

Na izbor baze podataka utiče i cena, i to ne samo cena platforme baza podataka već i cena potrebnog hardvera. Velike količine podatka zahtevaju dosta prostora na disku. Autori se u radu [16] bave osnovnim razlikama tradicionalnih *row oriented* i *column oriented* baza podataka i zaključuju da su za mnoge aplikacije koje zahtevaju analitičke sposobnosti baze podataka, *column oriented* sistemi bolji izbor jer imaju bolje karakteristike.

Performansama *row oriented* i *column oriented* skladišta se bave i autori u radu [17] upoređivanjem performansi komercijalne *row oriented* baze sa različitim konfiguracijama i *column oriented* baze. Utvrđeno je su performanse *row oriented* baza značajno slabije koristeći *Star Schema Benchmark (SSBM)*, kao i da postoje značajne razlike u performansama između ova dva sistema, kako na nivou sloja za izvršenje upita, tako i na sloju za skladištenje podataka. Navedeni benčmark predstavlja *Data Warehouse* benčmark, koji je izveden iz TPC-H benčmarka.

Iako je je generalno prisutno mišljenje da je za *Data Warehouse* i OLAP procese najbolje koristiti *column-store* sistem upisa podataka, autori su u radu [18] na osnovu istraživanja zaključili da se ovaj tip baza može koristiti i kod drugih aplikacija koje zahtevaju široke i slabo popunjene tabele, bez degradacije performansi. Imajući to u vidu, *column-store* baze se mogu koristiti za potrebe *Semantic Web* podataka, XML, kao i podataka sa GEM šemama.

Relacione baze podataka su optimizovane za I/O operacije, a pristup hijerarhiji keša na sistemu može imati veliki uticaj na performanse baze podataka. Komercijalne baze podataka koriste NSM (*N-ary Storage Modular Storage Model*) umesto DSM (*Decomposition Storage Model*) kao osnovni metod za skladištenje podataka iz razloga niže cene za rekonstrukciju podataka. Autoru su u radu [19] na osnovu izvršenih eksperimenata utvrdili da NSM model za skladištenje podatka može imati negativne efekte na performanse keš memorije.

Kada je potrebno izvršiti analitičke zadatke, baze podataka koje koriste *Column* orijentisano skladištenje podataka pokazuju bolje karakteristike performansi od tradicionalnih *Row* orijentisanih baza podataka. Razlog što *Column* orijentisane baze imaju mnogo bolje performanse kod operacija čitanja leži u tome što one pristupaju samo podacima kolona koje se koriste u upitima. Autori su u radu [20] izvršili eksperimente nad bazom HP Vertica koja koristi *column-store* zapise i MySQL baze podataka koja koristi *row store* zapise i predstavlja najkorišćeniju *Open Source* bazu podataka u upotrebi. Za potrebe merenja performansi oba sistema korišćeni su nasumično generisani podaci od nekoliko miliona redova. Rezultati izvedenih eksperimenata su pokazali da je HP Vertica imala znatno bolje performanse od transakcione MySQL baze kada se analitički upiti izvršavaju nad velikom količinom podataka. Razlika u performansama se smanjuje sa smanjenjem količine podataka.

Enorman rast podataka u današnjim sistemima korišćen od strane različitih aplikacija i socijalnih mreža uvodi *cloud computing* kao nezaobilazan faktor u obezbeđivanju kako skladišnog prostora, tako i obezbeđivanju dobrih performansi sistema koji su zaduženi za upravljanje podacima čija struktura postaje sve kompleksnija. Kada god razmatramo takve sisteme, procesi koji rade s podacima moraju biti visoko paralelni i da se izvršavaju nad različitim sistemima. Autori se u radu [21] bave ukupnim pregledom ove problematike uključujući analizu zahteva za današnje data generatore i pojašnjavaju dosadašnja rešenja, kao i probleme koji do trenutka pisanja rada nisu rešeni.

Optimizacija SQL upita ima značajnu ulogu za performanse sistema tradicionalnih relacionih baza podataka. Da bi se razumela uloga procesa optimizacije upita, autori su se u radu [22] osvrnuli kako na statičke tako i na dinamičke procese optimizacije upita. Takođe, razmatrali su i izvesna poboljšanja koristeći *Volcano* optimajzer koji se koristi kod objektno orijentisanih baza podataka.

Jedna od metoda za poboljšanje performansi sistema za upravljanje bazama podataka može biti i korišćenje metoda denormalizacije. Autori u radu [23] predlažu korišćenje denormalizacije kao međukoraka između logičkog i fizičkog modeliranja kod kreiranja analitičkih procedura za dizajn kriterijuma aplikativnih zahteva. Normalizacija predstavlja proces grupisanja atributa u predefinisanim strukturama podataka. Denormalizacija se može predstaviti kao proces za smanjenje stepena normalizacije radi povećanja performansi izvršenja upita. Jedna od najčešćih oblasti primene tehnika denormalizacije je u *Data Warehouse* implementacijama za *data mining* transakcije. Date smernice u ovom radu su generalne, tako da se mogu primeniti na najveći deo baza podataka. Poboljšanja u performansama se mogu dobiti samo ukoliko se smernice primene pod uslovom potpunog razumevanja aplikativnih zahteva.

Kod baza podataka postoji niz parametara koji mogu uticati na sveukupne performanse sistema. Jedan od tih parametara je *in-memory*. *In-memory* baze podataka čuvaju sve podatke u fizičkoj memoriji i zbog toga imaju veoma dobre karakteristike performansi i propusnog opsega transakcija u odnosu na baze koje upisuju podatke na čvrsti disk. Grupa autora je u radu [24] predstavila rezultate istraživanja merenja performansi ove dve vrste sistema sa odgovarajućom analizom, koristeći TPC-C benčmark.

Analizom performansi *Open Source* OLTP baza MySQL i MariaDB se bavila grupa autora u radu [25]. Rezultati istraživanja su pokazali da je u okruženju gde su obe baze podataka instalirane na Virtual Manager Virtual Machine Xen 4.4 i testirane koristeći dve vrste različitih opterećenja, MySQL prikazao bolje performanse u odnosu na MariaDB. Iz ugla korišćenja resursa obe baze podataka su koristile isti nivo opterećenja procesora i memorije.

3.2. Poređenje performansi NoSQL i relacionih sistema za upravljanje bazama podataka

Imajući u vidu sve nižu cenu skladištenja podataka kao i potrebu današnjih aplikacija da skladište i vrše upite nad velikom količinom podataka, primećuje se sve veća upotreba takozvanih NoSQL baza podataka, koje se koriste u konceptu *Big Data*. NoSQL baze su kreirane da pruže visoke performanse, u smislu brzine i skladištenog prostora, kao i visoke dostupnosti [26] [27] [28]. Može se reći da su one nus-proizvod web 2.0 ere potrebne dizajnerima web servisa sa velikim brojem korisnika. Danas postoji više od 120 ovakvih rešenja u ponudi.

Autori su u radu [29] izvršili poređenje NoSQL rešenja sa MySQL proizvodom koji je predstavnik relacione baze podataka, kako bi detektovali šta se dobija, a šta gubi korišćenjem NoSQL rešenja u odnosu na tradicionalno rešenje. Prvi preduzeti korak je izbor/klasifikacija proizvoda koji imaju sličnu svrhu i slične osobine. Selektovani NoSQL proizvodi koje su autori izabrali za poređenje sa MySQL su Cassandra i Hbase. Iz kvalitativnog ugla, autori su opisno prikazali poređenje sledećih karakteristika različitih proizvoda: postojanost, replikacija, visoka dostupnost, transakcije, svesnost o lokaciji podataka, podržani programski jezik, podrška različitih dobavljača i tip potrebne licence. Iz kvantitavnog ugla, autori su razmatrali poređenje karakteristika koje se odnose na aspekte količine skladištenja podataka i performansi.

Aspekt količine skladištenja podataka: Kako su NoSQL proizvodi dizajnirani za ogromne količine podataka ne postoji smisao njihovog poređenja sa MySQL bazom koja u proseku može da sadrži 1 milion zapisa bez memorijskog keširanja. Preko ovog limita rezultat upita postaje suviše spor.

Aspekt performansi: Korišćenjem laboratorijskog benčmarka koji koristi *YSCB (Yahoo! Cloud Serving Benchmark)* merenje je izvršeno na 120 miliona zapisa male veličine (1kB), na 6 noda, i 0.12 TB ekvivalentnih instalacija sva tri proizvoda, pri čemu su mereni sledeći parametri: *Read latency in a write intensive environment*, *Write latency in a write intensive environment*, *Read latency in a read intensive environment* i *Write latency in a read intensive environment*. Iako SQL i NoSQL baze podataka dele neke zajedničke mogućnosti, one se ipak ne ponašaju slično u datim okruženjima. To ukazuje da se one ne mogu koristiti kao zamenljive za rešavanje bilo koje vrste problema, već je potrebno napraviti odgovarajući izbor.

S obzirom na sve veću popularnost i korišćenje NoSQL baza podataka, veoma često se postavlja pitanje njihovog poređenja u smislu karakteristika i performansi sa standardnim relacionim bazama podataka. Veliki proizvođači relacionih baza podataka na tržištu kao što su Microsoft, IBM i Oracle su dugi niz godina ulagali značajna sredstva u njihov razvoj, a njihova

eksploatacija je prisutna u svim industrijama. Sa druge strane, NoSQL baze podataka su drugačije, nisu relacione, ne moraju da koriste *query* jezike, nemaju šemu, podržavaju jednostavnu i laku replikaciju i omogućavaju skladištenje ogromnih količina podataka. Glavna prednost NoSQL baza se ogleda u jednostavnom dizajnu, horizontalnoj skalabilnosti i finoj kontroli dostupnosti podataka. Cassandra je jedna od najpopularnijih NoSQL baza podataka i koristi se duže vreme za smeštanje podataka Facebook-a (razvijena je od strane Facebook-a). Autori su u radu [30] imali za cilj da uporede performanse SQL Servera i Cassandra rešenja, kako bi ustanovili u kom trenutku bi korisnici mogli da pređu sa relacione baze na NoSQL bazu i obrnuto. Za poređenje performansi obe baze korišćene su tabele sa po 1,000,000 redova na jednom nodu, a eksperiment je obuhvatao tri tipa operacija: *Read* nad poljem *Key* u SQL Serveru i *Key Value Pair* u Cassandra bazi; *Write* nad tabelama u obe baze; kao i operaciju *Select*. Cassandra podržava jednostavni *Select* upit, ali ima ograničenje da ne može da vrati više od 15,000 zapisa. Eksperiment je izveden na x64 računaru sa I3 procesorom, 4GB RAM i Microsoft Windows 10 Professional operativnom sistemom. Za Cassandra test je korišćena DataStax instalacija Cassandra verzije 3.4. Za SQL server test je korišćen SQL Server 2012. Test podaci su generisani koristeći AdventureWorks2012 bazu u SQL Serveru 2012, tačnije kreirane su tabele u obe baze koristeći SQL skriptu za punjenje 1,000,000 redova sa 26 kolona. Rezultati eksperimenata su pokazali da je prilikom uvoza podataka u bazu, SQL Server pokazao značajno bolje performanse u odnosu na Cassandra bazu. Cassandra bazi je bilo potrebno 321 sekundu dok je SQL Server završio taj proces za samo 34.42 sekunde. Kod izvoza podataka rezultat je bio sličan prethodnom, Cassandra je proces izvršila za 322 sekunde dok je SQL serveru trebalo samo 20.3 sekundi. Sličan odnos u rezultatima su pokazali i testovi sa *Select* upitima, pri čemu Cassandra ima limit u maksimalnom broju od 15,000 redova koje može da prikaže. Ograničenje izvršenih eksperimenata se ogleda u korišćenju samo jednog PC računara u test okruženju.

Uporedna studija između ne-relacione NoSQL MongoDB baze i relacione Microsoft SQL baze podataka pri obradi nestruktuiranih podataka u XML i JSON formatu je bila predmet istraživanja grupe autora u radu [31]. Fokus autora je bio u istraživanju mogućnosti koje pružaju obe baze podataka za slučaj skladištenja podataka koji nisu normalizovani. To je najčešći primer koji se koristi kod nestruktuiranih podataka socijalnih mreža. Istraživanje je bazirano na kreiranju benčmark aplikacije koju su autori razvili u C#, koristeći Visual Studio 2013. Imajući u vidu rad sa podacima koji ne mogu biti normalizovani, ali su i dalje povezani na neki način sa podacima u relacionoj bazi podataka, autori su izabrali hibridni model za SQL Server nasuprot modela za skladištenje podataka u MongoDB. U SQL serveru su skladišteni *ID* i *xml* tipovi podataka, dok su u MongoDB podaci skladišteni kao dokumenti. Cilj rada je da se odgovori na pitanje u kojim slučajevima je bolje koristiti relacione, a u kojim NoSQL baze podataka. Zaključak autora, na osnovu izvršenih eksperimenata je da u slučajevima kada nije moguće izvršiti normalizaciju podataka i u slučaju obrade velike količine podataka, MongoDB je bolje rešenje koja pruža značajno bolje performanse.

Istraživanje poređenja performansi relacionih i ne-relacionih baza podataka, konkretno Oracle i MongoDB kod izvršavanja kompleksnih upita nad velikom količinom podataka, koji su bili dostupni u dokument formatu i kao takvi konvertovani u Oracle tabele, su bili predmet rada autora u radu [32]. Za test podatke su korišćeni podaci sa adrese infochimps.com. Preko 500,000 redova iz većeg broja tabela u CSV formatu je uvezeno u Oracle i MongoDB. Za potrebe testiranja autori su pripremili nekoliko različitih tipova upita pri čemu je korišćen računar sa Intel® Core™2 Duo CPU T5750 @ 2.00GHz, 4 GB RAM, 40 GB particijom i softverima: Ubuntu 12.04 64bit, Oracle Database 11g Release 2, Mongo DB shell version: 2.0.9, Ubuntu 12.10 64-bit. Rezultati testiranja su pokazali da je MongoDB prikazao bolje performanse kod upita vraćanja rezultata, dok je podbacio u odnosu na Oracle kod izvršenja agregacionih funkcija. To dokazuje da NoSQL baze ne mogu biti zamena za relacione baze, već da u zavisnosti od zahteva koje treba ispuniti, treba izabrati odgovarajuće rešenje.

Grupa autora se u radu [33] takođe bavi poređenjem performansi NoSQL baze podataka MongoDB i relacione baze podataka Oracle. Radi najboljeg poređenja performansi obe baze podataka autori su razmatrali vreme izvršenja operacija nad bazom. Prvi test je uključivao operaciju ubacivanja podataka u obe baze podataka u sekvencama od po 10; 100; 1,000; 10,000; 100,000 i 1,000,000 redova. MongoDB-u je bilo potrebno znatno više vremena da ubaci manje setove podatka u odnosu na Oracle, ali je za veće setove podataka to radio znatno brže od Oracle. Slični testovi su urađeni i za operaciju ažuriranja podataka u bazi, gde je MongoDB pokazao znatno bolje karakteristike od Oracle, bez obzira na veličinu setova podataka. Sličan rezultat u korist MongoDB je dobijen i kod operacija brisanja podataka iz baze.

Autori su u radu [34] predstavili svoje istraživanje merenjem performansi nad relacionom bazom podataka MySQL i NoSQL bazom MongoDB. Eksperimenti su vršeni koristeći fizički računar sa Core i5, 8GB RAM i HD 500 GB. Za softversko okruženje su korišćeni Linux Mint 17.1 «Rebecca», Java «1.8.0_121», Apache Maven 3.2.5, YCSB 0.8.0, MySQL Server 5.5.54, i MongoDB 3.2.8. Nekoliko setova podataka od 1,000 do 600,000 redova su korišćeni za različite tipove i težine upita, kao i ažuriranja podataka u tabelama. Autori su primenili *Yahoo! Cloud Serving Benchmark* za potrebe arbitraže performansi ova dva sistema. Vreme izvršenja upita je bio glavni parametar koji su autori koristili za uporedno merenje performansi. Kod operacija čitanja podataka se pokazalo da je MySQL dobar izbor za manje do srednje velike baze podataka. Sa porastom količine podataka performanse MySQL sistema brzo opadaju. MongoDB je pokazao mnogo bolje karakteristike kod operacija čitanja u slučajevima velike količine podataka. Kod operacija upisivanja, MongoDB je pokazao značajno bolje karakteristike od MySQL. Jedan od razloga je i to što MongoDB ne primenjuje procese verifikacija kod upisivanja podataka, za razliku od MySQL. Ako se posmatra potreba za proširenjem baza podataka tokom eksploatacije i ukoliko se ne očekuje veliki porast količine podataka, MySQL može biti dobro rešenje. Međutim, ako su potrebe za značajnim proširenjem baza podataka vidljive, a ujedno se razmatra *Big Data* i *Cloud Computing* okruženje sa potrebom odziva u realnom vremenu, MongoDB predstavlja bolji izbor.

Slično istraživanje poređenja performansi MySQL i MongoDB su se bavili i autori u radu [35] pri čemu su za vršenje eksperimenata koristili realnu bazu podataka Supermarket upravljačkog sistema. Zaključak autora je da se obe baze podataka ponašaju slično u smislu vremena izvršenja upita kada se operacije vrše nad manjim setom podataka. Međutim, kako se količina podataka nad kojom treba izvršiti operacije povećava, MongoDB pokazuje bolje karakteristike performansi.

Autori se u radu [36] takođe bave poređenjem karakteristika MySQL i MongoDB. Sprovodeći niz eksperimenata dolaze do sličnih zaključaka kao u prethodnom radu. Autori predstavljaju i mogućnost integracije ove dve baze koristeći *middleware* komponente između njih, u slučajevima kada ovaj pristup može predstavljati najbolje rešenje.

Studija poređenja performansi MySQL i MongoDB, fokusirana na merenje parametara *threading* i sistemskih resursa (procesora i memorije) prilikom izvršenja operacija upisivanja i čitanja kompleksnih tipova podataka je predstavljena u radu [37]. MongoDB je u eksperimentima prikazao veću brzinu kod kompleksnih tipova i veće količine podataka, pri čemu je koristio više sistemskih resursa u odnosu na MySQL. Ovaj zaključak predstavlja bitan parametar u odlučivanju o izboru baze podataka, imajući u vidu cenu iznajmljivanja *Cloud* resursa potrebnih za MongoDB, dok je MySQL besplatan. Sa druge strane, MongoDB je lakši za korišćenje od strane programera u odnosu na MySQL, jer ne zahteva znanje SQL jezika, niti korišćenje normalizacije podataka. Pored toga, učitavanje i izvoz podataka je jednostavniji kod MongoDB.

Grupa autora je u radu [38] predstavila poređenje performansi MySQL i MongoDB na osnovu eksperimenata izvršenih nad manjom količinom podataka koje sadrže do 30,000 redova. PHP *Open Source* programski jezik je korišćen za spajanje podataka iz dve tabele koristeći MySQL *join* upit i identičnu agregacijsku funkciju *inner join* u MongoDB. Mereni su parametri brzine odgovora (*response time*) i vremena vraćanja podataka (*retrieval time*). Eksperimenti su vršeni na jednom fizičkom računaru sa Intel I5 procesorom. Rezultati eksperimenata su pokazali da je u datim slučajevima MySQL pokazao bolje performanse u odnosu na MongoDB, što pokazuje da je kod manje količine podataka MySQL bolji izbor, ne samo po pitanju performansi, već i cene.

Poređenjem performansi MongoDB i MySQL baze podataka se bavio i autor u radu [39]. On je vršio ispitivanja metrika nad procesima učitavanja manje i veće količine podataka kod obe baze podataka. Pri tome je koristio *Yahoo! Cloud Serving Benchmark (YCSB)* i tri tipa testa, od ukupno šest koja ovaj benčmark definiše i to: A- *Heavy update* (50% operacija čitanja, 50% operacija ažuriranja), C- *Read only* (100% operacija čitanja) i F – *Read-Modify-Writes* (50% operacija čitanja, 50% operacije čitanja, upisa, ažuriranja). Testovi su vršeni nad virtualnim mašinama kreiranim na Amazon *Cloud* platformi. Na osnovu izvršenih eksperimenata autor

zaključuje da je MongoDB pokazao bolje performanse od MySQL za sva tri tipa testa koja uključuju rad sa većom količinom podataka.

Graph NoSQL baze podataka sve više dobijaju na popularnosti zbog brzine obrade kompleksnih upita. Ove baze podataka koriste *traversal based* jezik upita poznat kao *Cypher*, koji omogućava korisnicima da pretražuju šablone i veze među nestruktuiranim podacima. *Graph* baze koriste *graph* strukture, nodove i ivice (*edges*), dok relacione baze podataka koriste tabele i relacije među njima. Autori su u radu [40] predstavili rezultate merenja performansi *Graph* baze Neo4j i relacione baze podataka MySQL. Za potrebe eksperimenata su korišćeni različiti tipovi upita koje se mogu svrstati u 4 kategorije: selekcija-pretraga, rekurzija, agregacija i pronalaženje šablona u podacima nad realnom bazom podataka „Career Village“. Rezultati testiranja su pokazali da Neo4j ima bolje performanse u odnosu na MySQL u najvećem broju slučajeva, a posebno u slučajevima pronalaženja šablona u podacima i rekurzivnim upitima kod operacija sa velikom količinom podataka.

Studijom merenja performansi dokument NoSQL baze CouchDB i relacione baze podataka MySQL se bavila grupa autora u radu [41]. Procena performansi je bazirana na korišćenju CRUD operacija (*CREATE*, *READ*, *UPDATE*, *DELETE*) nad obe baze podataka. Glavni cilj studije se ogledao u izradi komparativne analize uticaja obe baza podataka na performanse aplikacije koja pokreće CRUD operacije. Za potrebe testiranja autori su kreirali dve slične aplikacije razvijene u Javi, pri čemu je svaka koristila odgovarajuću bazu podataka. Dve strukture podataka su kreirane za CouchDB, prva u kojoj je svaki dokument sadržao referencu ka drugom dokumentu i drugu strukturu u kome je svaki dokument sadržao sve podatke o svakom entitetu. Da bi se dobili relevantni rezultati svaki upit/operacija je primenjena na opsegu od 1,000 do 1,000,000 elemenata. Prva struktura u CouchDB je pokazala manju efikasnost u odnosu na drugu, bez obzira na broj elemenata. MySQL baza se pokazala kao efikasnija u slučajevima manjeg broja elemenata i pored toga što indeksi nisu kreirani nad bilo kojom tabelom. Sa aspekta performansi, druga struktura CouchDB je pokazala bolje performanse u odnosu na MySQL kod *insert*, *select* i *delete* operacija, pogotovo kod operacija sa velikom količinom podataka.

Kada dođe do pojave novih baza podataka, kreiraju se novi benčmark testovi i vrše testiranja performansi sa ranijim bazama podataka. Najčešće se dešava da rezultati merenja performansi pokazuju da su baze novijeg datuma brže od tradicionalnih. Grupa autora je u radu [42], na osnovu istraživanja utvrdila da ova poređenja performansi ne moraju uvek biti tačna. Oni su ponovo razmotrili neka ranija istraživanja performansi Neo4j graph baze podataka i relacionih baza podataka i došli do mogućih uzroka koji su doveli do ne-fer zaključaka o mnogo lošijim performansama testiranih relacionih baza podataka. Autori su otkrili da su rezultati nekih testova koji su izvršeni na PostgreSQL bazi pokazali veliku zavisnost od faktora kao što su konfiguracija, šema baze podataka, fino podešavanje i slično. Štaviše, autori su na osnovu testiranja dobili dobre rezultate performansi za takva scenarija testiranja koja su ranijim

istraživanjima i testiranjima prikazana kao loša za relacione baze podataka. Preporuka autora je da svaki sistem koji se testira treba fino podesiti za odgovarajuće testove, pa čak i kreirati sopstvene benčmarke kako bi se dobili fer rezultati.

Današnje potrebe za skladištenjem velike količine podataka i frekventnim operacijama nad velikim setovima podataka su sve veće. Grupa autora je u radu [43] prikazala studiju poređenja performansi dve NoSQL baze podataka, Cassandra i Hbase i relacione SQL baze podataka Citus/PostgreSQL. Sve tri baze podataka su testirane u okruženju klastera *Single Board* kompjutera (SBC). *Single Board* kompjuter je kompjuter smešten na jednoj matičnoj ploči sa mikroprocesorom, memorijom i drugim hardverskim komponentama koje ima tradicionalni računar. Klaster se sastojao od 15 *Raspberry Pi* 3 nodova sa *Docker Swarm* orkestracionim alatom za primenu servisa i balanserom opterećenja nad *Single Board* kompjuterima. Pretpostavka autora je da ovakav klaster može reprezentovati *cloud* servis okruženje uključujući skalabilnost, elastičnost i visoku dostupnost. Autori su koristili *Yahoo Cloud Serving Benchmark (YCSB)* koji predstavlja benčmark za *cloud* baze podataka i sastoji se od nekoliko različitih testova opterećenja. Rezultati testiranja su pokazali da postoji kompromis između parametara performansi i replikacije. Oba parametra su značajna kada se uzima u obzir konfiguracija distribuiranih sistema koristeći SBC. Cassandra je postigla bolje rezultate koristeći nivo izdržljivosti podesive od strane korisnika, dok su Citus i Hbase pokazale dobru izdržljivost, ali sa padom performansi u slučaju povećanja broja replika.

Poređenjem parametara performansi NoSQL i SQL baza podataka za operacije čitanja, upisivanja i brisanja na *key-value* tipu skladišta NoSQL baza i tradicionalnih SQL baza bio je predmet rada autora u radu [44]. Generalno gledano NoSQL baze podataka su optimizovane za skladištenje podataka koje koriste *key-value* sistem, dok tradicionalne SQL baze podataka to nisu.

Najboljim izborom baze podataka koja bi skladištila podatke Geografskog Informacionog sistema (GIS) se bavila grupa autora u radu [45]. Količine podataka u kojoj korisnici na internetu označavaju geo podatke stalno rastu. Sa druge strane aplikacije koje koriste geo podatke imaju potrebe za većim propusnim opsezima. U okviru ovog istraživanja autori su izvršili eksperimente nad relacionom bazom PostgreSQL i dve NoSQL baze MongoDB dokument baze i Neo4j graph baze. Za potrebna testiranja korišćeni su OpenStreetMap podaci. Sve tri baze su instalirane na 64-bit Windows okruženju sa Intel® core™ i7-7700 HQ procesorom na 2.80 GHz i 16 GB RAM memorije. Autori su kreirali dva jednostavna test upita za selekciju svih korisnika koji su označili neko mesto na mapi i za selekciju svih nodova kreiranih od strane korisnika. Rezultati testiranja su pokazali da je MongoDB pokazao najbolje karakteristike performansi u datoj postavci.

Tradicionalne relacione baze podataka se sve češće porede sa novijim sistemima baza podataka koje mogu da upravljaju ogromnim količinama podataka i koje se generalno mogu podeliti na

NoSQL, NewSQL i sisteme bazirane na pretrazi (*Search based systems*). NewSQL sistemi predstavljaju inovirane relacione baza podataka koje mogu da ponude slične karakteristike performansi kao NoSQL baze za *Online Transaction Processing (OLTP)* operacije čitanja i pisanja, pri čemu obezbeđuju glavne karakteristike specifične tradicionalnim relacionim bazama podataka kao što su ACID (*atomicity, consistency, isolation, durability*). ACID predstavljaju svojstva transakcija baza podataka koja obezbeđuju validnost podataka bez obzira na greške i anomalije koje se mogu pojaviti. Grupa autora je u radu [46] izvršila analizu arhitekture, karakteristika i klasifikacija NewSQL baza podataka za OLTP transakcije. Takođe je u radu dat i pregled poredivih metrika tradicionalnih, NoSQL i New SQL baza. Svako rešenje ima svojih prednosti i mana. Ako posmatramo metrike performansi, NoSQL i NewSQL imaju bolje karakteristike, ali i veće propuste u bezbednosnom aspektu u odnosu na tradicionalne relacione baze podataka.

Korišćenje virtualizacionih rešenja nastavlja da se uvećava u informacionim sistemima. Vezom između performansi baza podataka i sistema virtualizacije koja se može koristiti, tačnije uticajem virtualizacije tipa I na performanse NewSQL baze podataka se bavi autor u radu [47]. Razlika između Tipa I i II virtualizacije je u tome što se tip I izvršava direktno na hardveru (*bare-metal*) dok se kod tipa II virtualizacija izvršava na operativnom sistemu. Isti hardver je korišćen za testiranje performansi okruženja bez i sa virtualizacionim softverom i to: Hewlett Packard Enterprise (HPE) Apollo r2600 šasija sa 3 HPE dual-processor noda. Svaki procesor se sastojao od dva Intel Xeon Broadwell E5-2698v4 2.4GHz CPU, pri čemu je svaki imao 14 korova i 50MB keša, 128GB DDR4-2400 memorije i jedan 480MB SATA *solid-state drive* (SSD). Komunikacija između nodova se odvijala pomoću 1 GbE *Ethernet* portova. Za okruženje sa virtualizacijom je korišćen VMware ESXi virtualizacioni softver. Merene su sledeće metrike: korišćenje procesora, memorije, disk i mrežni ulaz/izlaz kao i broj transakcija u sekundi iniciranih od strane baze podataka. Svaki nod je imao isti operativni sistem CentOS 7.7.1908, *Open Source* verziju Red Hat Enterprise Linux. Dva noda su korišćena za klaster baze podataka NuoDB NewSQL relacione baze podataka, na kojima se nalazio i softver za merenje parametara performansi. Treći nod je korišćen za pokretanje skripti opterećenja. Treba napomenuti i da je NuoDB baza podataka dizajnirana za rad na *bare-metal* konfiguraciji, virtualizacionom i *cloud* okruženju. Rezultati testiranja su pokazali da je stepen opterećenja procesora uvećan za preko 60% kod okruženja sa virtualizacijom. Korišćenje memorije je diskretno uvećano kod okruženja sa virtualizacijom u opsegu između 1.4% i 2.7%. Disk I/O operacije su pokazale uvećane vrednosti kod virtualizacionog okruženja za više od 30%. Navedeni rezultati direktno ukazuju da virtualizaciono okruženje prouzrokuje dodatno opterećenje na sistem.

Grupa autora je u radu [48] je predstavila podršku JSON *data management* operacijama u verziji Oracle baze 12cR1 koja omogućava korisnicima da skladište, indeksiraju podatke i vrše upite koristeći JSON data model. Na taj način je omogućen rad sa podacima u relacionoj bazi podataka bez prethodnog definisanja šeme, što je karakteristika NoSQL baza podataka. Autori

su takođe predstavili binarni format upita za dekodiranje JSON data modela koji smanjuje razliku u performansama između šema kodiranih relacionih podataka i JSON tekstualnih podataka bez šeme, koje se koriste u NoSQL bazama. Sa dodavanjem ovih novih mogućnosti Oracle relaciona baza podataka je pozicionirana kao dobro rešenje, kako za slučajeve koji zahtevaju fiksnu SQL šemu, tako i za fleksibilnu šemu koja se koristi kod NoSQL pristupa.

Ako posmatramo *Data Warehouse* rešenja za potrebe izveštavanja koristeći sve veće količine podataka kojim današnje poslovanje raspolaže, možemo zaključiti da ona mogu biti veoma kompleksna i da im je za rad potrebno sve više vremena. Autori su u radu [49] izvršili poređenje i analizu performansi baza podataka za *Big Data Warehouse* potrebe. Testirane su dve baze podataka, MySQL klaster koji pripada tradicionalnoj SQL bazi podataka i Hive nad Hadoop-om koji pripada NoSQL bazi podataka. Za potrebe testiranja autori su pripremili nekoliko različitih scenarija opterećenja. Za procenu performansi je korišćen *Star Schema Benchmark (SSB)*. Rezultati eksperimenata su pokazali slabe performanse MySQL klastera za velike setove podataka i pored mogućnosti korišćenja dodatnih slobodnih resursa, što dovodi do zaključka da je ovo rešenje najbolje koristiti za transakcione sisteme. Hive je sa druge strane pokazao skalabilne rezultate, pogotovo kada je u mogućnosti da koristi dodatne resurse, što ga pozicionira kao dobro rešenje za ogromne količine podataka.

Koristeći postojeća istraživanja drugih autora koja su bila dostupna do trenutka objave rada [50], grupa autora je izvršila poređenje karakteristika SQL i NoSQL baza podataka, poređenjem 10 različitih karakteristika, počevši od cene i skalabilnosti do koegzistentnosti i bezbednosti. Zaključak autora je da tradicionalne relacione baze podataka predstavljaju najbolji izbor u slučajevima korišćenja limitirane količine struktuiranih podataka za razliku od NoSQL baza podataka, koje karakteriše velika skalabilnost i dobre performanse kod operacija nad velikom količinom podataka, sa osvrtom na ograničene mogućnosti sa aspekta bezbednosti.

Izborom odgovarajućih rešenja koristeći istraživanja drugih autora u segmentu poređenja karakteristika SQL i NoSQL baza podataka kao i poređenjem *Cloud database* sistema se bave autori u radu [51]. Na osnovu predstavljenih analiza autori preporučuju današnjem poslovanju sledeća rešenja:

- NoSQL baze podataka su dobar izbor za najveći deo najzahtevnijih operacija koje operišu sa ogromnim količinama podataka. One su brze, skalabilne i pouzdane.
- Amazon RDS (MySQL) nudi više konfiguracionih opcija, dobre performanse i dobra su preporuka za većinu organizacija.
- Manje firme bi trebalo da koriste Amazon SimpleDB zbog dobrih karakteristika sa manjom količinom podataka i povoljne cene
- *Firebase Realtime* baze podataka (NoSQL) su najbolji izbor za ličnu upotrebu ili za grupu korisnika, pogotovo kada nije potrebno pristupiti bazi podataka po principu 24/7 i kada je potrebno koristiti jednostavno skladište podataka.

U današnjem poslovanju efikasnost informacionih sistema predstavlja osnovni preduslov i najčešće se meri vremenom vraćanja rezultata upita, brojem konkurentskih korisničkih sesija koje može da obradi, kao i nivoom energetske efikasnosti sistema. Na optimizaciju informacionog sistema se veoma često gleda kao na kompleksan i neophodan zadatak. Porastom količine podataka može doći do pada performansi sistema i obično se na sistem baza podataka gleda kao na sistem čije karakteristike performansi treba izmeriti, a na osnovu rezultata iste unaprediti. Autor u radu [52] daje sistematičan pregled analize 117 različitih istraživanja koje su se bavile performansama različitih tradicionalnih relacionih i NoSQL baza podataka.

Detaljnijom analizom izdate literature koje se bave poređenjem softverske arhitekture, karakteristika i performansi SQL i NoSQL baza podataka se bavi i grupa autora u radu [53]. Zaključak autora na osnovu istraživanja je da nema potrebe za tranzicijom tradicionalnih relacionih podataka na NoSQL baze podataka. Svaka baza podataka ima svojih prednosti i mana, te da treba napraviti odgovarajući izbor na osnovu zahteva koje treba ispuniti. Generalno posmatrano, ako organizacija treba da zadovolji zahteve za standardizacijom i koegzistencijom podataka onda je najbolji izbor relaciona baza podataka. NoSQL baze su najbolji izbor kod velike količine nestruktuiranih podataka.

U trenutku objave rada [54], preko 200 različitih NoSQL baza podataka je bilo raspoloživo za korišćenje obezbeđujući različite karakteristike i funkcionalnosti. Autori su u ovom radu predstavili istraživanje merenja performansi nekoliko najčešće korišćenih *in-memory* NoSQL baza podataka: MongoDB, Redis, Memcached, Cassandra i H2 relacione baze podataka. H2 se razlikuje od ostalih relacionih baza podataka po tome što čuva podatke u memoriji umesto na hard disku. Za potrebe testiranja performansi navedenih baza autori su razvili specijalizovani softver u Javi za različite eksperimente i to: upisivanje parova ključeva, čitanje vrednosti za odgovarajući ključ, brisanje parova ključeva i učitavanje svih podataka. Svi eksperimenti su vršeni na istom fizičkom računaru sa Intel i7 procesorom i 16GB RAM. Testovi su vršeni nad bazama podataka sa različitim brojevima zapisa, od 1,000 do 1,000,000. Zaključak autora je da SQL i NoSQL baze podataka poseduju različite karakteristike i da one nisu zamena jedne za drugu. Ukoliko je zahtev za koegzistentnost podataka u bazi nefleksibilan, onda je relaciona baza podataka najbolji izbor. U suprotnom, NoSQL baze podataka su bolji izbor zbog obezbeđivanja visoke dostupnosti, skalabilnosti i visokih performansi.

3.3. Poređenje performansi različitih NoSQL sistema za upravljanje bazama podataka

Povećanjem količine podataka na Internetu kao i na osnovu zahteva web 2.0 baziranih aplikacija NoSQL baze podataka postaju opšte prihvatljiv izbor za skladištenje i upravljanje podacima. Autori u radu [55] predstavljaju data modele NoSQL baza podataka, mogućnosti upita koji se mogu nad njima postavljati, mehanizme za kontrolu konkurentnog pristupa podacima, načine particionisanja podataka koji se distribuiraju na višestruke nodove, kao i replikacione karakteristike. Imajući u vidu sve veći izbor NoSQL baza podataka koje nude različite karakteristike i mogućnosti, uvek je potrebno izvršiti odgovarajuću procenu najboljeg rešenja shodno zahtevima koji se trebaju ispuniti.

Grupa autora se u radu [56] bavi razlozima za tranziciju sa tradicionalnih relacionih podataka na NoSQL baze podataka i analizira performanse tri najpopularnija NoSQL rešenja: Cassandra, MongoDB i CouchBase. Cassandra je predstavnik *Key-Value* baze, optimizovana je za operacije upisivanja podataka, nema centralni nod i podaci se mogu čitati ili upisivati na bilo koji nod u klasteru. Rezistentna je na pad bilo kog noda u klasteru time što će drugi nod preuzeti opterećenje. Karakteriše je horizontalna skalabilnost. Cassandra je Apache 2.0 licencirano rešenje. MongoDB je dokument orijentisana NoSQL baza podataka koja ne koristi šemu i podatke upisuje u BSON/JSON binarnom formatu, koji pruža mogućnost jednostavne integracije podataka sa najčešće korišćenim aplikacijama. Karakteriše je horizontalna skalabilnost i otporna je na pad bilo kog noda u klasteru. MongoDB poseduje centralni (*master*) nod i višestruke potčinjene (*slave*) nodove. Ova baza podataka je *Open Source* rešenje licencirano od strane kompanije 10.gen. CouchBase je *Open Source* dokument orijentisana NoSQL baza podataka. Namenjena je radu sa interaktivnim web i mobilnim aplikacijama. Podaci se skladište u JSON formatu i baza je otporna na pad pojedinačnih nodova. Karakteriše je horizontalna skalabilnost i licencirana je od strane Apache 2.0. Na osnovu različitih testova opterećenja nad sve tri baze, rezultati eksperimenata su pokazali da MongoDB čita podatke iz baze brže od Cassandra baze, ali sporije od CouchBase. Cassandra i CouchBase brže upisuju podatke u odnosu na MongoDB. Opšti zaključak je da izbor jednog rešenja nije uvek jednostavan proces, imajući u vidu da je svako od navedenih rešenja optimizovano za različite tipove opterećenja i zahteva.

Poređenjem performansi 5 NoSQL baza podataka: Redis, *Key-value store*, verzije 3.2.1.; MongoDB *Document Store* verzije 3.2.7; CouchBase *Document Store*, verzije 4.0.0; Cassandra *Column family Store*, verzije 2.2.7 i HBase *Column family Store*, verzije 0.98.20 su se bavili autori u radu [57]. Korišćen je *Yahoo! Cloud Serving Benchmark (YCSB)* koji radi u dve faze, učitavanje podataka i testiranje transakcija. Prilikom testiranja se može fino podešavati težina

opterećenja koristeći odnos pisanja i čitanja, kao i definisanje opsega podataka u fazi testiranja. Autori su izabrali tri standardna testa opterećenja ovog benčmarka, označena slovnim oznakama A,C i H nad slučajno generisanim podacima u svim bazama podataka. Zaključak izveden na osnovu rezultata testova ukazuje da je Redis najbolje pozicioniran kod učitavanja podataka i izvršenja operacija nad podacima, mada su vidljiva ograničenja u radu sa ekstremno velikom količinom podataka. *Document* i *column-family* NoSQL baze su pokazale dobre prosečne performanse.

Detaljnijom studijom poređenja karakteristika i performansi NoSQL baza podataka se bavila grupa autora u radu [58]. Tačnije, tri NoSQL baze podataka su bile predmet istraživanja i to CouchBase, MongoDB i RethinkDB koristeći dva scenarija, jednostrukih i višestrukih niti (threads). Rezultati eksperimenata su pokazali da CouchBase ima bolje performanse u skoro svim izvršenim operacijama, izuzev kod operacija učitavanja i vraćanja skupa dokumenata sa višestrukim nitima, gde je MongoDB pokazao bolje rezultate.

Postoje različiti primeri u mnogobrojnim industrijama kao i na internetu koji zahtevaju skladištenje i rad sa slikama, kao što su analiza saobraćaja, medicinski zapisi, socijalne mreže i dr. Grupa autora je u radu [59] predstavila istraživanje poređenja performansi dokument orijentisanih NoSQL baza, MongoDB i CouchBase u slučajevima upisivanja i čitanja slika različitih veličina iz baze podataka. Kod jednostavnih operacija čitanja i pisanja slika manjih data setova nije utvrđena velika razlika. U slučajevima operacija upisa veće količine slika, MongoDB je zahtevao više vremena u odnosu na CouchBase, dok je kod operacija čitanja slika rezultat bio suprotan, Mongo DB je radio brže u odnosu na CouchBase.

NoSQL baze podataka obezbeđuju otpornost na pad nodova u klasteru kao i visoku dostupnost. Radi obezbeđivanja ovih osobina koriste se replikacioni mehanizmi. Autori su u radu [60] predstavili studiju uticaja replikacije na performanse NoSQL baza podataka. NoSQL baze podataka koriste odgovarajuće protokole za ispunjenje koegzistentnosti/doslednosti (*consistency*) podataka u procesu replikacije na druge nodove u klasteru. Ovi protokoli kontrolišu kako se podaci prenose između nodova, kako se rukuje operacijama čitanja i pisanja, kada se kopije podataka na drugim nodovima ažuriraju, kao i koji je zadovoljavajući nivo koegzistencije podataka. Kod mehanizma replikacije koriste se sledeći parametri: replikacioni faktor, parametar koji definiše broj nodova na koji se jedan podatak replicira i nivo koegzistencije, koji definiše koliki broj replika/nodova mora da odgovori na dolazeći upit kako bi se smatrao validnim. Ovi parametri mogu značajno da utiču na performanse rada baze podataka. Niži nivoi koegzistentnosti dovodi do boljih performansi i obrnuto. Dve NoSQL baze su bile u fokusu ove studije, Cassandra i MongoDB. Autori su koristili YCSB benčmark za izvođenje eksperimenata nad okruženjem koje je postavljeno na *Private Cloud* infrastrukturi. Korišćeni su testovi visokog opterećenja čitanja i pisanja. Rezultati eksperimenata su pokazali da je sa aspekta propustnosti podataka kod operacija intenzivnog čitanja MongoDB pokazao za nijansu bolje rezultate od Cassandra baze, dok je kod operacija intenzivnog pisanja Cassandra

nadjačala performanse Mongo DB za više od 70 procenata. Na taj način je potvrđeno da je Cassandra optimizovana za operacije intenzivnog upisa podataka. Sa aspekta uticaja replikacije na performanse, utvrđeno je da replikacioni model *Master-Slave* MongoDB ima manji uticaj na performanse propusnosti podataka u odnosu na *Multi-Master* replikacioni model Cassandra baze.

Kod NoSQL baza podataka, parametri performansi, dostupnosti i koegzistentnosti podataka su veoma povezani. Da bi utvrdili nivo povezanosti ovih parametara autori su u radu [61] predstavili istraživanje uticaja različitih podešavanja koegzistentnosti na skalabilnost, latentnost (kašnjenje) i propusni opseg (*throughput*) tri replicirana Cassandra klaster okruženja na Amazon Web Servisu (AWS), podešena prema različitim scenarijima. Rezultati istraživanja su potvrdili prethodna očekivanja autora, da povećana koegzistentnost podataka rezultira smanjenjem propusnog opsega baze i povećanjem kašnjenja operacija čitanja i pisanja. Najbolje performanse za sva postavljena podešavanja koegzistentnosti se dobijaju kada se Cassandra klaster nalazi u jednom data centru.

Ista grupa autora je istraživala performanse operacija čitanja i pisanja na Cassandra klasteru koji je postavljen na Amazon EC2 Cloud-u [62]. Jedan od najznačajnijih saznanja dobijen eksperimentima je da je moguće smanjiti kašnjenja u operacijama Cassandra baze optimalnim podešavanjima parametra koegzistentnosti za obe operacije, čitanja i pisanja. Visoki parametri koegzistentnosti podataka utiču na pad performansi za oko 25%, dok najbolja podešavanja koegzistentnosti zavise od odnosa operacija čitanja i pisanja. Autori u svom radu predlažu metodologiju benčmarka za optimizaciju podešavanja koegzistentnosti, kako bi se dobile maksimalne performanse Cassandra baze, a ujedno i obezbedila dobra koegzistentnost podataka kod obe vrste operacija.

Grupa autora je u radu [63] predstavila istraživanje uticaja uloge aplikacionog *timeout-a* (vremenskog ograničenja) kao glavne determinante u relaciji između systemske dostupnosti i performansi koje se ogledaju u odzivu servisa. Aplikacioni *timeout* predstavlja glavni pokazatelj grešaka ili preopterećenosti koje se mogu javiti u radu servisa i upravo je on inicijator operacija za oporavak u slučaju otkaza. Arhitekta sistema bi trebale da imaju mogućnost podešavanja željenog vremena odgovora sistema, vodeći računa o odgovarajućem odnosu između koegzistentnosti podataka i njihove dostupnosti. Međutim, mnogi distribuirani sistemi kojima pripada i Cassandra koriste statička *timeout* podešavanja koja često mogu biti previše velika, dovodeći u pitanje efikasnu detekciju grešaka i inicijaciju mehanizama za upravljanje otkazima. Preporuka autora na osnovu izvršenih eksperimenata je da optimalni *timeout* treba da bude dinamički i da zavisi od strukture baze, veličine i odnosa operacija čitanja i pisanja. Takođe, potrebno je uključiti i faktore kao što su trenutno opterećenje sistema, broj replika kao i parametre koegzistentnosti.

Istraživanje NoSQL baza podataka, MongoDB (*document store*), Cassandra (*column-store*) i Redis (*key-value store*) sa fokusom na poređenje njihovih performansi koristeći YCSB benčmark je bio predmet studije autora u radu [64]. Izvršeno je testiranje navedenih baza podataka koristeći šest različitih tipova opterećenja nad setovima podataka različite veličine. Na osnovu izvršenih eksperimenata Redis je prikazao najbolje performanse čitanja u odnosu na ostale dve baze. Razlog ovakvih performansi Redis-a se ogleda u mogućnosti korišćenja promenljive memorije za smeštanje i vraćanje podataka. MongoDB je bio brži u operacijama čitanja od Cassandra baze. Sa druge strane, MongoDB je pokazao bolje performanse u operacijama skeniranja od Redis-a i Cassandra baze.

Sa porastom korišćenja različitih servisa i aplikacija na Internetu vidljiv je i porast multimedijalne komunikacije korisnika koristeći mobilne uređaje. Potreba za multimedijalnom komunikacijom je vidljiva u svim industrijama kao i na aplikacijama socijalnih mreža. Za obezbeđivanje i kontrolu mrežne multimedijalne komunikacije se koristi *IP Multimedia Subsystem* (IMS) arhitektura, a u okviru nje *Session Initiation Protocol* (SIP) koji upravlja uspostavljanjem, održavanjem i prekidanjem multimedijalnih poziva između dva uređaja. Mnoštvo parametara multimedijalnih sesija SIP protokol smešta u bazu podataka koja može biti lokalna, na istom serveru gde se nalazi i SIP servis. Kako je multimedijalna komunikacija servis u realnom vremenu, SIP zahteva visoke performanse svih zavisnih komponenti, kako ne bi došlo do degradacije samog servisa. Iz tog razloga su *In-memory* NoSQL baze podataka alternativno rešenje za brži rad sa podacima, koji se mogu čuvati u memoriji. Grupa autora je u radu [65] predstavila studiju poređenja performansi SIP servera (OpenSIPS) koristeći tri *key-value* NoSQL baze podataka, lokalnu bazu na OpenSIPS serveru, Memcached i Redis. U testovima su vršena merenja parametara memorije, propusnog opsega i vreme odgovora na poziv. Rezultati testiranja su pokazali da lokalna baza na OpenSIPS serveru koristi manje memorije u odnosu na druge dve baze. Sa aspekta propusnog opsega najbolji rezultat je prikazala takođe lokalna baza. Zaključak autora je da je lokalna baza na OpenSIPS serveru najbolji izbor iz perspektive performansi, ukoliko se ne zahteva da se podaci čuvaju u memoriji duže vremena nakon zatvaranje sesije koja ih je kreirala. Ukoliko se ova karakteristika zahteva, onda je Memcached baza podataka odličan izbor.

Godinama unazad su relacione baze podataka bile izbor za smeštanje podataka kliničkih istraživanja u oblasti zdravstva. Međutim, zbog veće količine podataka i sve većom potrebom za boljim performansama, skalabilnošću, visokom dostupnošću NoSQL baze podataka su postale predmet razmatranja kao potencijalnog rešenja. Grupa autora je u radu [66] predstavila Oracle NoSQL bazu podataka kao izbor rešenja u svojoj studiji.

Grupa autora je u radu [67] predstavila istraživanje merenja performansi Hbase i Cassandra NoSQL baza podataka, koristeći Java bazirani alat YCSB za pokretanje različitih testova opterećenja na setovima podataka od 100,000 do 250,000 zapisa. Za potrebe testiranja i analize je korišćen *Yahoo Cloud Servicing Benchmark*. Okruženje obe baze podataka je postavljeno na

virtuelnoj instanci sa 2 VCPU i 8GB memorije. Za izabrane testove čitanja i ažuriranja podataka u bazi, Hbase je pokazao bolje performanse u odnosu na Cassandra.

Na izbor odgovarajuće NoSQL baze podataka za neku aplikaciju utiču mnogi parametri, od kojih jedan od najznačajnijih mogu biti performanse kod izvođenja takozvanih CRUD operacija (*CREATE, READ, UPDATE, DELETE*). Grupa autora je u radu [68] predstavila istraživanje dve najčešće korišćene dokument NoSQL baze podataka, MongoDB i dokument MySQL u smislu izmerenih performansi pri izvođenju CRUD operacija. Obe baze podataka su dobar izbor za *Big Data* aplikacije sa odličnim vremenom izvršenja operacija, bez obzira na njihovu kompleksnost i veličinu obuhvaćenih setova podataka. Prednost dokument MySQL-a se ogleda u jednostavnosti i lakšem korišćenju, pogotovo za one korisnike koji imaju znanja iz oblasti relacionih baza podataka.

Merenje i poređenje performansi MongoDB i CouchDB baza podataka je bilo predmet istraživanja autora prezentovanog u radu [69]. Zaključak studije je da ukoliko baza podataka treba ispuniti uslov veće efikasnosti i brzine rada, onda je MongoDB bolji izbor u odnosu na CouchDB. Međutim, u slučaju da je potrebno obezbediti multi master replikaciju baze, kao i podršku za mobilne uređaje onda je CouchDB bolji izbor.

Uporedna analiza poređenja performansi šest NoSQL baza podataka je predstavljena u radu [70]. Autori su koristili *Yahoo! Cloud Serving Benchmark* radi izvedbe eksperimenata nad MongoDB, Cassandra, HBase, Redis, CouchBase i OrientDB bazama podataka. Testiranja su vršena koristeći šest različitih tipova opterećenja definisana navedenim benčmarkom i različitim setovima podataka na fizičkom računaru sa Core i5, 8GB RAM, HD 1 TB i sledećim softverskim okruženjima: Ubuntu 16.04, 64-bit, MongoDB 3.2.8, CouchBase 2.5.2, Cassandra 2.2.5, HBase 0.94.8, Redis 3.0.2 i OrientDB 2.1.3. Neki od ključnih zaključaka studije su da je CouchBase pokazao najveću brzinu učitavanja velikih setova podataka u odnosu na druge baze podataka. Hbase je demonstrirao najbolje performanse kod upisivanja novih zapisa u bazu, dok su Cassandra i Hbase demonstrirali najbolje performanse ažuriranja postojećih zapisa u odnosu na ostala rešenja.

3.4. Pregled, mogućnosti i performanse ETL alata

Data Warehouse ili DW predstavlja analitičku bazu podataka (OLAP - *Online Analytical Processing*) koja se koristi za potrebe poslovne analitike, utvrđivanje trendova i izveštavanja, koristeći različite izvore podataka. Glavni cilj DW je pružanje podrške poslovnim procesima, analizama i procesima donošenja poslovnih odluka na osnovu konsolidovanih istorijskih podataka [71]. Dizajniranje DW sistema predstavlja kompleksan projekat i zahteva dobro poznavanje kako poslovnih procesa i podataka vezanih za te procese, tako i tehnička znanja potrebna za izradu rešenja koje će ispuniti očekivanja korisnika i opravdati investiciju [72]. *Business Intelligence* ili BI predstavlja skup tehnologija, alata, procesa i praksi koje pomažu organizacijama da analiziraju i pretvore podatke u korisne informacije. Cilj poslovne inteligencije je omogućavanje donosiocima odluka u organizacijama da steknu dublji uvid u poslovanje i donesu najbolje odluke na osnovu analize istorijskih podataka i dobro dizajniranog BI rešenja [73] [74] [75].

Jedan od važnih procesa pri kreiranju naprednog analitičkog rešenja je punjenje DW sistema podacima iz različitih izvora podataka. Izvori podataka najčešće imaju različite strukture i različite šeme, te ih je potrebno transformisati kako bi odgovarali potrebnoj šemi i strukturi DW sistema. Podaci na izvoru mogu biti upitni sa aspekta kvaliteta, te se često dešava da neki atributi nedostaju, da neki nisu koegzistentni i slično, te je pre punjenja DW baze potrebno primeniti tehnike i procese "čišćenja" podataka. Pored toga, podaci se u produkcionoj transakcionoj bazi koje predstavlja izvor podataka konstantno ažuriraju i dopunjuju, te je potrebno konstantno podatke u DW sistemu sinhronizovati sa podacima u bazi izvora. Navedeni izazovi se rešavaju adekvatnim procesima i alatima koji se pokreću u odgovarajućim intervalima i nazivaju se ETL (*Extraction-Transformation-Loading*) procesima i alatima. ETL proces u osnovi predstavlja skup zadataka (*tasks*) koji vrše ekstrakciju podataka iz izvorne baze, transformaciju, čišćenje i učitavanje podataka u DW bazu [76].

Iako se DW rešenja koriste od početka 1990-ih godina, dizajn i modeliranje ETL procesa se i dalje izvršava u alatima koji su zavisni od platforme i šeme u kojoj se nalazi destinaciona baza podataka. Autor je u radu [77] predstavio prvi modeling pristup za specifikaciju ETL procesa na način koji nije zavistan od dobavljača DW baze i koji automatizuje generisanje odgovarajućeg programskog koda na komercijalnim ETL platformama.

Od kvaliteta podataka zavisi i kvalitet analitičkog rešenja. Grupa autora je u radu [78] predstavila istraživanje najčešće korišćenih alata za "čišćenje" podataka. Oni su sve izazove vezane za kvalitet podataka podelili na tzv. probleme na nivou šeme podataka i na nivou instance. Problemi koji postoje na nivou šeme se mogu rešiti poboljšanjem dizajna šeme i integracijom podataka, dok problemi na nivou instance mogu biti povezani sa zasebnim

zapisima ili višestrukim zapisima. Najčešće se to dešava u slučajevima kao što su nedostatak validnog podatka u *Not Null* polju (slučaj kada se polje popunjava nasumično, kako bi se izbeglo pojavljivanje grešaka prilikom upisa zapisa u bazu) ili sintaksne greške operatera koji unosi podatak u polje i slično. U zavisnosti od tipa zadataka koje raspoloživi alati obavljaju, autori su ih podelili u šest grupa: alati za profilisanje podataka, alati za analizu podataka, alati za transformaciju podataka, alati za čišćenje podataka, alati za eliminisanje duplikata i alati za obogaćivanje podataka. Analizom i pregledom ETL alata se bavila i grupa autora u radu [79], predstavljajući uporednu studiju komercijalnih alata i njihove karakteristike u odnosu na dimenzije kvaliteta podataka, kao što su duplikacija, kompletnost, tačnost, konciznost, doslednost. Najčešće korišćeni komercijalni alati predstavljeni ovoj studiji su: Talend Data Integration, Pentaho Data Integration, Microsoft SQL Server Integration Services i Informatica Data Integration.

Postoji značajan broj aplikacija koje su razvijene za izvršenje ETL procesa. Izbor najboljeg alata za odgovarajuće zahteve je uvek izazov i izbor zavisi od njegove efikasnosti, performansi, cene i drugih karakteristika. Autori su u radu [80] predstavili studiju uporedne analize ETL procesa koristeći dva ETL alata, Microsoft SQL Server 2008 Integration Service (SSIS) i Pentaho Data Integration (PDI). Za izvor podataka je korišćena Northwind baza. Scenario za izvođenje eksperimenata na SQL serveru se sastojao od:

- ekstrakcije podataka iz Northwind baze postavljene na SQL serveru,
- procesa transformacije podataka, kako bi se ispunili dati zahtevi za učitavanje podataka u DW bazu na SQL serveru i
- učitavanje podataka u DW koristeći data import funkcionalnost SQL servera.

Scenario za izvođenje eksperimenata koristeći Pentaho se sastojao od:

- ekstrakcije podataka iz Northwind baze postavljene na SQL serveru,
- procesa transformacije podataka u MySQL format podataka, kako bi se ispunili dati zahtevi za učitavanje podataka u DW bazu na SQL serveru i
- učitavanje podataka u DW koristeći data import funkcionalnost Pentaho Data Integration.

Rezultati eksperimenata su pokazali da je integracija podataka koristeći SSIS komplikovanija i vremenski zahtevnija u odnosu na PDI, te da je u PDI bolji izbor.

Još jedna grupa autora se bavila poređenjem karakteristika Microsoft SQL Server Integration Service (SSIS) i Pentaho Data Integration (PDI) ETL alata [81]. Oba alata predstavljaju robusna rešenja. SSIS nudi više mogućnosti kroz konfiguracije objekata mada je kod kompleksnijih operacija potrebno izvršiti kodiranje. Prednost i snaga SSIS leži u jednostavnim kontrolnim tokovima, tokovima podataka i arhitekturi koja je bazirana na događajima, što programerima nudi visoku fleksibilnost dizajna ETL procesa. PDI sa druge strane nudi više opcija za

povezivanje i uvoz podataka iz Google Analytics, kao i dodatne opcije za pristup veb servisima. Može se koristiti kako na Windows, tako i na Linux operativnim sistemima. Generalno posmatrano, SSIS zadovoljava potrebe velikih organizacija i radi sa velikim količinama podataka. Ukoliko posmatramo *Open Source/besplatne* alate, PDI se može koristiti kod manjih organizacija i u tim slučajevima može pokazati dobre performanse.

Najčešći je slučaj da sve moderne baze podataka imaju ugrađene alate koje omogućavaju brzi izvoz i uvoz podataka u bazu. Grupa autora je u radu [82] izvršila istraživanje poređenja performansi tih alata kod nekoliko baza podataka koje se koriste na tržištu i to: HP Vertica, MariaDB Columnstore, Oracle i MS SQL Server. Za potrebe izvođenja eksperimenata korišćene su tabele sa podacima Vmart šeme. Vmart je *Data Warehouse Data Mart* koji se koristi kao primer šeme u Vertica bazi podataka. Korišćeno je ukupno 7 tabela sa podacima u rasponu od 10,000 do 5,000,000 redova u tabeli. Eksperimenti su izvedeni na Windows 10 host računaru sa Intel I7-6700HQ procesorom na 2.60 GHz i VMware virtuelnim mašinama (jedna virtualna mašina po bazi podataka). Svakoj VM je dodeljeno po 6GB RAM. Za potrebe kreiranja tabela i relacija Vmart šeme, na svakoj virtuelnoj mašini je korišćenja SQL skripta sa odgovarajućim komandama. Za svaku bazu podataka u navedenoj studiji pripremljene su i izvršene skripte za uklanjanje i uspostavljanje ograničenja podataka (*data constraints*), kao što su “*Check*”, “*Foreign Key*”, “*Primary Key*”. Cilj upotreba ovih skripti je da mereno vreme za izvršenje testiranih operacija ne uključi i izvođenje operacije ograničenja podataka. Takođe, za potrebe testiranja performansi na nivou svake baze podataka su pripremljene skripte za testove koje izvoze podatke iz tabela u CSV fajlove, kao i skripte koje uvoze podatke iz CSV skripti u bazu podataka. Na osnovu rezultata izvršenih testova na nivou različitih platformi, utvrđena je vrlo mala razlika u brzini izvođenja operacija izvoza podataka iz baze u fajl i uvoza podataka iz fajla u bazu podataka koristeći ugrađene alate. Ipak, najbolje vreme punjenja baze sa podacima iz najvećih tabela je postigao Oracle SQL Loader, dok je najbolje vreme izvoza podataka postignut naredbom “*Copy*” u HP Vertica.

DW koncept se ogleda u skladištenju ekstrahovanih, agregiranih i filtriranih podataka u meta-bazama podataka koje pružaju mogućnost rada sa višeslojnim, višedimenzionalnim skupovima podataka neophodnim za izvršenje odgovarajućih analiza i donošenje odluka. Za potrebe studije, grupa autora je u radu [83] koristila OLTP bazu podataka “Data Learning System” (DLS) originalno razvijenu od strane autora za potrebe srednje tehničke škole, kako bi pripremila odgovarajuće ETL procese za učitavanje podataka u DW bazu podataka. Autori su dizajnirali ETL proces koji se sastojao od pripreme (brisanje ograničenja spoljnih ključeva u DW bazi, brisanje svih podataka DW baze), popunjavanja DW dimenzionih tabela podacima iz DLS baze i drugih izvora, popunjavanja DW tabela činjenica podacima iz DLS baze i drugih izvora, i na kraju vraćanja ograničenja spoljnih ključeva u DW tabele. Za izvođenje ETL procesa korišćen je SQL Server Integration Services. Zaključak autora je da se predstavljeno ETL rešenje može primeniti kod OLTP baza podataka koje ne sadrže veliku količinu podataka

i kada postoji potreba da se prvo uklone svi podaci iz DW baze podataka, a zatim u DW izvrši učitavanje novih podataka.

Grupa autora je u radu [84] izvršila poređenje različitih karakteristika popularnih ETL alata i to Informatica, Datastage, AbInitio, Oracle Data Integrator i Microsoft SSIS i izvršila analizu njihovih prednosti i mana. Takođe, autori su analizirali i neke dodatne karakteristike kao što su optimizacija performansi, analiza podataka u realnom vremenu i predstavili ih u okviru svoje studije. Slična analiza karakteristika različitih ETL alata, uključujući pored nekih od prethodno navedenih i IBM Information Server, Talend Open Studio, SAS Data Integration Studio, Pentaho i Clover ETL objavljena je u radu [85].

Pogled na odgovarajući izbor i implementaciju ETL procesa pri izradi DW rešenja, koristeći realnu studiju slučaja za agenciju za poljoprivredu Indonezije su predstavili autori u radu [86].

Iako je ETL nezaobilazan alat u projektima izrade DW, potrebno je imati na umu da nisu svi ETL softverski alati isti. Najbolji ETL alat treba izabrati na osnovu komponenti okruženja koje učestvuju u projektu i primerima slučaja za koje će biti korišćeni. Grupa autora je u radu novijeg datuma [87] predstavila pregled i komparativnu analizu dostupnih ETL alata. ETL alati su podeljeni po kategorijama i zajedničkim karakteristikama, što ih čini lakšim za odabir na osnovu zahteva koje trebaju ispuniti. ETL alati su predstavljeni iz ugla industrije ili tipa organizacije za koje se mogu primenjivati, kao i za najoptimalniju veličinu organizacije. Na kraju, na izbor odgovarajućeg alata utiče i njegova cena te su autori u svojoj analizi uključili i taj aspekt.

U *Big Data* eri zadatak ETL alata je da procesiraju velike količine nestruktuiranih i struktuiranih podataka radi kompleksnih analiza poslovanja. Za standardne ETL alate je to veliki izazov, te je razmatranje njihovog unapređenja neophodno radi opravdanja investicije poslovanja. Jedna od ideja je dizajniranje uređaja sa niskim budžetnim komponentama koje karakterišu adekvatni kompjuterski i skladišni parametri za izvođenje ETL procesa. Grupa autora u radu [88] predlaže Hadoop na iSCSI preko *Ethernet adapted Network Attached Storage* (NAS) kao rešenje za izvođenje ETL operacija na osnovu studije poređenja performansi ekstrakcije podataka, performansi transformacije i performansi učitavanja podataka sa standardnim HDFS fajl sistemom (*Hadoop Distributed File System*). Na osnovu rezultata eksperimenata, predložena arhitektura sa niskim budžetnim komponentama je pokazala skalabilne performanse za operacije sa podacima u *Big Data* segmentu.

ETL predstavlja najzahtevniju komponentu analitičkog rešenja i prema autorima u publikaciji [89] na ETL odlazi od 60% do 80% vremena kompletnog DW i BI projekta. Sa porastom broja izvora podataka i količine podataka standardni pristup u radu sa ETL alatima koji su se zasnivali na dodavanju hardvera, kako bi se dobile bolje performanse, više nije opcija. Grupa autora je u svojoj studiji [90] predstavila poređenje performansi komercijalnih ETL alata i Hadoop alata. Hadoop je *Open Source* Apache projekat. Analiza dobijenih rezultata performansi je pokazala

da su Hadoop rešenja bolja u poređenju sa komercijalnim ETL alatima za rad sa velikom količinom podataka.

Poređenje ETL tehnika predstavlja veoma zahtevan zadatak kod koga postoje brojni izazovi koje treba imati u vidu, kako bi se dobili što tačniji rezultati. Autori u radu [91] predstavljaju uporednu studiju različitih ETL procesa sa aspekta poznatih ETL izazova, kao i izazova ETL tehnika testiranja.

Tradicionalni ETL alati se najčešće izvršavaju na jednom serveru i u slučaju geo prostornih podataka takav pristup nije u stanju da obezbedi dobre performanse, pogotovo kada su u opseg uključene velike količine geo prostornih podataka. Za unapređenje performansi ETL procesa se može koristiti *MapReduce* bazirana paralelizacija [92]. Grupa autora je u radu [93] predstavila istraživanje performansi ETL procesa velike količine geo prostornih podataka prikupljenih od strane *Internet of Things* (IoT) senzora i predložila D_ELT (*delayed extracting-loading – transforming*) proces korišćenjem *MapReduce* paralelizacije. IoT senzori generišu velike količine podataka koji se moraju brzo analizirati, jer vrlo često stariji podaci nemaju veliku upotrebnu vrednost. Brzina pripreme podataka je u ovim slučajevima ključna za dobijanje dobrih rezultata analize. Korišćeni eksperimenti su poslužili za merenje i poređenje performansi standardnih ETL/ELT (*extracting-loading-transforming*) pristupa i D_ELT pristupa, koristeći različite količine podataka i nivoa kompleksnosti. Rezultati eksperimenata su pokazali da je D_ELT pristup imao bolje performanse u odnosu na ETL ili ELT, pogotovo kod obrade veće količine i kompleksnosti podataka, zbog većeg stepena paralelizacije.

U današnjem *Big Data* svetu koji se karakteriše sa 4V (*Variety, Velocity, Volume and Veracity*), odnosno karakteristikama raznolikosti, brzine, obima i verodostojnosti, brzina rada sa velikim količinama podataka predstavlja osnovni imperativ svakog rešenja. ETL rešenje mora da ima sposobnost rada u *Cloud* infrastrukturi, koju odlikuje enormna snaga procesiranja i ogroman skladišni prostor. Najveći izazov u ovakvom okruženju je cena, obzirom na poslovni model plaćanja po potrošnji. Autori su u radu [94] predstavili istraživanja karakteristika performansi različitih alata i predložili smernice za izbor odgovarajućih rešenja, imajući u vidu odnos performansi i cena.

Radi unapređenja performansi ETL procesa se mogu koristiti odgovarajuće strategije koje uključuju optimizaciju procesa ekstrakcije podataka, smanjenje redundantnosti podataka i korišćenje paralelizacije [95]. Za poboljšanje performansi ETL procesa za potrebe *Data Warehouse*, grupa autora je u radu [96] predložila tehniku keširanja upita i njegovih rezultata, kako bi se u slučaju postavljanja sličnog upita od strane drugih korisnika koristili keširani rezultati.

Poznato je da se za potrebe analitičkih funkcija nad podacima, pored CPU koristi i GPU (*graphics processing unit*). CPU izvršava sve zahteve instaliranog softvera na računaru. GPU može podržati CPU izvršavajući konkurentne kalkulacije i može pojedine zadatke podeliti u

nekoliko manjih i izvršiti ih u paraleli. Razmatranje uključivanja GPU u ETL procese radi povećanja performansi bilo je predmet studije [97], u kojoj je grupa autora na osnovu izvršenih eksperimenata procesiranja setova podataka različitih veličina, različitih tipova upita i različitih tipova opterećenja, zaključila da se najbolje karakteristike performansi mogu dobiti adekvatnim distribuiranjem opterećenja na kombinovano CPU-GPU okruženje.

Imajući u vidu ulogu ETL procesa u izradi analitičkih rešenja, predmet istraživanja grupe autora objavljenog u radu [98] je pronalaženje modela koji bi na jednostavniji način predstavio ETL procese. Oni su pojednostavljeni pristup predstavili koristeći tri kategorije: modeliranje na osnovu mapiranja odgovarajućih izraza, modeliranje konceptualnih konstrukcija i modeliranje na osnovu UML (*Unified Modeling Language*) okruženja. Druga grupa autora, čiji je fokus takođe bio na modeliranju ETL procesa, je u radu [99] analizirala različite ETL pristupe modeliranja, počevši od tradicionalnih do onih koji se koriste u sadašnjem *Big Data* okruženju, poredeći njihove karakteristike. Studija prikazuje osnovne izazove vezane za dizajn *Big Data Warehouse* rešenja, kao i nedostatak generičkog dizajn modela za prikupljanje podataka, njihovo smeštanje, procesiranje, generisanje upita i analizu.

Kompletna pregled osnovnih principa, ETL tehnika i najboljih praksi je dat u radu [100]. Ovaj rad uključuje više oblasti, počevši od provere kvaliteta podataka, testiranja, optimizacije pa do bezbednosti i performansi ETL-a. Najznačajnije najbolje prakse uključuju bitnost mapiranja podataka i njihovu validaciju za vreme procesa integracije, kontrolu kvaliteta podataka tokom procesa transformacije, particionisanje i indeksiranje podataka radi efikasnog punjenja baze, definisanje i upravljanje rasporedom aktivnosti, kao i rukovanje greškama. Kvalitet definisanih procesa utiče između ostalog i na povećanje performansi kompletnog rešenja.

3.5. Pregled, primena i karakteristike BI alata

Business Intelligence (BI) je skup tehnologija, alata, procesa i praksi koje pomažu organizacijama da prikupe, obrade, analiziraju i pretvore podatke u korisne informacije. Cilj BI je omogućiti donosiocima odluka u organizacijama da steknu dublji uvid u poslovanje putem analize velike količine podataka, radi donošenja najboljih odluka za poslovanje. Ključni elementi BI uključuju skladištenje podataka, alate za analizu podataka, alate za izveštavanje i vizualizaciju, kao i tehnike prediktivne analize.

Značaj i proces donošenja odluka u današnjem poslovanju predstavlja kompleksan proces i oslanja se na izbor jedne od više dostupnih alternativa. Najveći izazov je izbor najoptimalnije alternative koja ima za cilj unapređenje poslovnih rezultata. Autori u radu [101] koriste tri dimenzije modernog pristupa donošenja poslovnih odluka: kvalitativan aspekt, kvantitavan aspekt kao i informativno-komunikacioni aspekt. Eksperimenti su izvršeni nad tri različita tipa podataka iz različitih izvora koristeći BI metode i tehnike. Rezultati su predstavljeni u tabelarnom obliku za prikaz kvantitavnog aspekta (kroz opis rezultata korišćenjem atributa i pod-atributa), za prikaz kvalitativnog aspekta i vizuelnim predstavljanjem dobijenih podataka za prikaz informaciono-komunikacionog aspekta. Autori su u svom istraživanju koristili AHP metodu analize (*analytical-hierarchical process*) donošenja odluka, koja je bazirana na konceptu ravnoteže za određivanje ukupnog relativnog značaja grupe atributa, aktivnosti ili kriterijuma.

BI sistemi su dugi niz godina predstavljali jedan od najbitnijih prioriteta poslovnog rukovodstva. Donosioci odluka u BI projektima rade zajedno sa dizajnerima rešenja kako bi rešenje pružilo relevantne informacije za prave poslovne odluke. Većina preduzeća danas poseduje neko BI rešenje. Nakon investicije i implementacije BI rešenja, donosioci poslovnih odluka često vrše analizu rezultata svojih odluka, a na osnovu informacija dobijenih od strane implementiranih rešenja. Autor u radu [102] predstavlja istraživanje direktnih i indirektnih efekata upravljanja BI rešenjem na kvalitet donesenih odluka od strane poslovnih rukovodioca. On koristi model parcijalnih najmanjih kvadrata (PLS - Partial Least Square Modelling) za analizu odgovora koje su viši IT menadžeri u Australiji dostavili u anketi. Rezultati utvrđuju ukupan efekat, a takođe otkrivaju posredne efekte kvaliteta podataka i opsega BI rešenja.

Grupa autora u radu [103] proučava odnos između BI funkcija, direktne koristi podrške u odlučivanju i koristi organizacije u kontekstu okruženja odlučivanja. Istraživanje koristi kvantitativnu studiju koja je zasnovana na anketi, a modelom parcijalnih najmanjih kvadrata analizira odgovore 228 firmi koje pripadaju različitim industrijama u regionu bliskog istoka. Rezultati studije potvrđuju očekivanu pretpostavku da postoji smisleni odnos između funkcija poslovne inteligencije, podrške u poslovnom odlučivanju i koristi za poslovnu organizaciju.

Kvalitet BI rešenja uveliko zavisi od kvaliteta podrške donosioca odluka u poslovnom preduzeću. Grupa autora je u svom istraživanju [104] verifikovala značaj uloge donosioca odluka u implementaciji BI rešenja, na praktičnom primeru globalne kompanije koja se bavi logistikom. U ovom primeru je razmatrana podrška dobijena od strane različitih menadžment nivoa i uspešnost BI rešenja, koja su u direktnoj sprezi. Rezultat podrške menadžmenta BI inicijativama dovodi do boljeg iskorišćenja podataka radi ostvarivanja konkurentske prednosti.

BI alati i rešenja pronalaze primenu u svakoj industriji. Planiranje potreba i efikasnost lanca snabdevanja je od ključnog značaja u farmaceutske proizvodnji. Grupa autora u svom radu [105] predstavlja okvir za predviđanje poslovanja. On se sastoji od specijalizovanih alata i različitih koraka koji mogu da pomognu donosiocima odluka da tačno predvide prodaju i zadovoljstvo korisnika, a na taj način i performanse firme. Predloženi okvir se sastoji od niza komponenti i uključuje korišćenje odgovarajućeg softvera za generisanje optimalnog predviđanja, kao i Power BI alata za praćenje tačnosti dobijenih predviđanja prodaje.

Analiza BI tehnologija u eri *Big Data* na primeru Alibaba grupe, pionira u oblasti e-trgovine [106] istražuje transformacionu ulogu poslovne inteligencije. Opseg ovog istraživanja uključuje BI tehnologije poslednje generacije koje su u primeni, pri čemu je jedan od glavnih fokusa na inteligentne sisteme preporuka, personalizovane marketing strategije kao i efikasno upravljanje lancem snabdevanja. Analizom dostupnih podataka se kreiraju prilagođene preporuke za ponudu proizvoda korisnicima platforme, koje imaju za cilj kako povećanje prodaje, tako i veće zadovoljstvo korisnika. Marketinške strategije koje se baziraju na podacima, omogućavaju Alibabi da kreira personalizovane prodajne promocije i lojalnost korisnika. Pravovremenom isporukom proizvoda upravlja kompleksan sistem inteligentnog upravljanja lancem snabdevanja, koji u realnom vremenu prati odgovarajuće aktivnosti optimizacije transporta. Studija slučaja "Singles' Day Global Shopping Festival" prikazuje snagu poslovne inteligencije u obezbeđivanju jedinstvenog dinamičnog doživljaja kupovine za krajnjeg korisnika.

Istraživanje primene poslovne inteligencije u malim i srednjim preduzećima u oblastima upravljanja lancima snabdevanja je predstavljeno u radu [107]. Izazovi u ovoj oblasti nastaju zbog ograničenja resursa, kao i zbog operativnih ograničenja. Grupa autora u svojoj studiji predlaže odgovarajući model za povećanje efikasnosti upravljanja lancima snabdevanja u malim i srednjim preduzećima, strateškom primenom poslovne inteligencije. Osnovni podaci su dobijeni kroz polustrukturirane intervjuje koji su zatim analizirani odgovarajućim tematskim analizama. Rezultati analize ukazuju na okvir koji se sastoji od 98 početnih tema koje su analizom podeljene na 55 osnovnih tema, 15 tema koje se odnose na organizaciju i 4 globalne teme koje obuhvataju nekoliko različitih analitika, počevši od analitike mreže snabdevanja, analitike poslovnog učinka, do optimalnog upravljanja putanjom i analizom zadržavanja lojalnosti.

Poslovna inteligencija nalazi veliku primenu u obrazovanju. Primer oblasti primene poslovne inteligencije u visokoškolskim ustanovama je predmet studije grupe autora u radu [108]. Ove ustanove donose odluke u nekoliko različitih domena, veoma često bez korišćenja podataka koje bi podržale potrebne odluke, dovodeći u pitanje ispravnost takvih odluka. Na taj način se može ugroziti uspeh i efikasnost obrazovne ustanove. Ovi izazovi se mogu rešiti korišćenjem poslovne inteligencije koja se bazira na odgovarajućim podacima. Predstavljena studija prezentuje sistematičan pregled literature koja predlaže odgovarajuću mapu puta za implementaciju sistema poslovne inteligencije u visokoškolskim ustanovama.

Radi odgovarajućeg usvajanja i primene BI u poslovnom preduzeću, kao i ostvarivanja postavljenih poslovnih ciljeva, potrebno je prethodno prepoznati faktore koji na to utiču [109]. Grupa autora u ovoj studiji koristi karakteristike organizacije, njenu kompleksnost, korisničku percepciju, tehnologiju, znanje i društveni okvir za identifikaciju ključnih aspekata. Koristeći odgovarajuće analize primarnih podataka, poslovna preduzeća mogu dobiti koristan uvid za rešavanje izazova primene poslovne inteligencije.

Grupa autora u svom istraživanju [110] istražuje faktore koji utiču na korišćenje BI u bankarskoj industriji. Poseban fokus predstavljenog istraživanja se odnosi na uticaj radnog iskustva zaposlenih na korišćenje BI u komercijalnim bankama u Jordanu. Model analize je razvijen korišćenjem TOE (Technology Organization Environment) okvira radi procene uticaja različitih faktora na korišćenje BI u bankama. Kreirana je empirijska analiza i dizajniran upitnik za prikupljanje podataka od strane zaposlenih. Grupa autora je analizirala 277 odgovora korišćenjem parcijalnog modeliranja najmanjih kvadrata, radi testiranja studije. Rezultati istraživanja ukazuju pozitivan uticaj faktora TOE na upotrebu poslovne inteligencije u bankama.

Pregled trendova u alatima za poslovnu inteligenciju u finansijskom sektoru sa posebnim osvrtom na njihovu upotrebu u SAD i Africi [111], pokazuje kako se finansijska industrija razvija. Korišćenje podataka i poslovne inteligencije predstavlja osnovni preduslov za održivi rast. Ova studija se bavi ispitivanjem ključnih indikatora razvoja i izazova koji oblikuju implementaciju BI rešenja na oba kontinenta. U SAD je primetan porast korišćenja BI u finansijskom sektoru, vođen sve većom potrebom za analitikom podataka u realnom vremenu, kao i usklađenosti sa regulativama. Kroz integraciju veštačke inteligencije i mašinskog učenja u BI alate obezbeđuje se napredna prediktivna analitika koja omogućava poslovnom rukovodstvu da brzo donosi odgovarajuće odluke. Mogućnost korišćenja BI u oblaku donosi mnoge prednosti kao što su dostupnost, skalabilnost i pristupačnost. U afričkom finansijskom sektoru je vidljiv napredak u korišćenju BI rešenja. Međutim, kontinent koji karakterišu različiti nivoi razvijenosti ekonomija se suočava se jedinstvenim izazovima. Vlade zemalja na afričkom kontinentu kao i njihovi finansijski sektori ulažu u odgovarajuću tehnološku infrastrukturu koja treba da ispuni odgovarajuće preduslove za dalja unapređenja na ovom polju.

Imajući u vidu današnja poslovanja koja su bogata podacima, sposobnost izvlačenja odgovarajućih uvida iz ogromne količine tekstualnih i struktuiranih podataka predstavlja jedan od najbitnijih ciljeva BI. Grupa autora u svojoj studiji [112] predstavlja istraživanje integracije analitike teksta, veštačke inteligencije i rešenja za planiranje resursa radi poboljšanja rezultata poslovne inteligencije. Koristeći algoritme veštačke inteligencije nad tekstualnim podacima dodatno se olakšava njihova analiza, pri čemu se mogu detektovati odgovarajući obrasci i trendovi koji doprinose boljem donošenju odluka. Kroz integraciju sa sistemom za planiranje resursa se može dobiti holistički pogled na poslovanje poslovnog preduzeća.

Razvoj tehnologija mašinskog učenja i integracija sa alatima poslovne inteligencije predstavlja dodatni potencijal za donošenje pravih poslovnih odluka [113]. Razumevanje i predviđanje učinka zaposlenih u poslovnim preduzećima predstavlja ključan faktor uspeha u odeljenjima zaduženim za efikasno upravljanje ljudskim resursima. Koristeći prediktivno modeliranje i analizu istorijskih podataka, današnje preduzeća mogu da identifikuju obrasce i moguće ishode, kako bi na najefikasniji način upravljali talentima u svojoj organizaciji.

Istraživanje uticaja veštačke inteligencije, analitike i velike količine podataka u oblasti zdravstva, konkretno na kvalitet nege pacijenata i izbor destinacije u oblasti medicinskog turizma je bio predmet istraživanja grupe autora [114]. Prezentovana studija se bazira na višestrukim parametrima koje obuhvata poslovna inteligencija kao što su zadovoljstvo pacijenata, izbor destinacije, ishodima lečenja, konkurentnost, isplativost i drugo.

Studija primene sistema donošenja odluka na komercijalnoj berzi koristeći poslovnu inteligenciju i upravljanje na primeru konvolucionih neuronskih mreža (*Convolutional Neural Networks* - CNN) je predstavila grupa autora u svom radu [115]. Istraživanje predstavljeno u ovoj studiji se bazira na transformacionom uticaju mašinskog učenja sa posebnim fokusom na konvolucione neuronske mreže, kao što su VGG16, ResNet50 i InceptionV3, na poljima organizacionog menadžmenta i poslovne inteligencije. Ovo istraživanje uključuje korišćenje sveobuhvatne metodologije, pri čemu se posebno naglašava korišćenje mašinskog učenja radi boljih rezultata u donošenju poslovnih odluka. Rezultati studije su pokazali veoma dobre performanse CNN modela u odnosu na tradicionalne algoritme, dostižući stopu tačnosti od čak 89.45%.

Prethodni primeri pokazuju široku upotrebu BI rešenja u različitim industrijama. Na tržištu su dostupni mnogobrojni BI alati i često postavljeno pitanje je koji alat predstavlja najbolji izbor za konkretne potrebe koje definiše poslovanje. Pored funkcionalnosti i performansi koje BI alati pružaju treba razmotriti i njihovu cenu, posebno imajuću u vidu da su neki od njih besplatno dostupni, a neki su podložni licenciranju. Grupa autora u svojoj studiji predstavlja i poredi BI alate koji se najčešće koriste u kompanijama u Turskoj [116]. Studija predstavlja listu BI alata koji se najviše reklamiraju, pri čemu Microsoft Power BI i SAP zauzimaju prva dva mesta. Data je i detaljna analiza funkcionalnosti i karakteristika, kako različitih verzija BI alata

istog proizvođača, tako i uporedna analiza mnogobrojnih karakteristika različitih BI alata i cena. Komparativnom analizom karakteristika različitih BI alata se bave i grupe autora u radovima [117], [118], [119], [120] i [121].

Benčmark analiza BI alata otvorenog koda koji se najčešće koriste u zdravstvu je predmet studije grupe autora [122]. Osnovni cilj studije je analiza besplatnih alata i mogućnosti primene u zdravstvenim klinikama, imajući u vidu sve specifičnosti i zahteve kliničkog okruženja. Autori su analizirali karakteristike šest BI alata: QlikView, Palo BI Suite, Jaspersoft BI, Tableau Public, Spago BI i Pentaho BI Suite. Na osnovu sveobuhvatne procene karakteristika svih alata prema definisanim zahtevima, zaključeno je da je alat Spago BI najbolji izbor.

4. Poređenje brzina izvršenja upita nad velikim količinama podataka koristeći različite sisteme za upravljanje bazama podataka

Količine podataka koje se generišu u današnjoj eri socijalnih mreža, internet i mobilnih aplikacija, kao i svim industrijama su enormne. Radi unapređenja poslovanja, kompanije vrlo često koriste baze podataka instalirane u svojoj internoj poslovnoj mreži, kao i kod različitih *cloud* provajdera na internetu, kako bi iz njih dobili poslovno vredne informacije radi donošenja poslovnih odluka. Tradicionalne baze podataka koje se koriste za smeštanje velike količine podataka, a koje karakterišu velike brzine konkurentnih procesa upisivanja u skladište se nazivaju OLTP baze podataka ili Online Transactional Processing SUBP. Međutim, ovaj tip baza podataka ne poseduje dobre karakteristike performansi kada je potrebno iz skladišta podataka generisati rezultate složenih upita. Za potrebe brzog generisanja *ad-hoc* izveštaja koristeći velike količine podataka, bazu podataka treba da karakteriše velika brzina čitanja. Pored toga, da bi se dobile bolje karakteristike performansi izvršenja upita, potrebno je prethodno izvršiti denormalizaciju tabela iz baze. Takođe, treba kreirati i manji broj tabela koje će sadržati sve potrebne informacije za one tipove upita koji su potrebni za generisanje izveštaja koje očekuje poslovanje.

Najveći broj sistema za upravljanje bazama podataka (SUBP) koristi arhitekturu skladišta podataka u kojima se podaci upisuju po redovima u svakom ciklusu upisa (*row-oriented data storage*). Na taj način se dobijaju visoke brzine pisanja. Pored ove arhitekture, postoji i arhitektura skladišta podataka koja vrši upisivanje podataka po kolonama, koja za rezultat ima značajno bolje performanse čitanja. Obzirom da se upiti zasnivaju na selekciji sadržaja kolona na osnovu filtera postavljenih u samom upitu, baza sa rednim upisom podataka mora da pročita sve kolone iz svih selektovanih tabela, te da iz njih filtrira odgovarajuće podatke iz traženih kolona. Kod baza podataka sa upisom po kolonama je taj proces mnogo brži, jer su podaci već upisani po kolonama i čita se samo sadržaj potrebnih kolona.

Da bi se dobile najbolje performanse sistema za poslovnu analitiku i izveštavanje, kod organizacija sa velikom količinom podataka najbolji pristup je koristiti baze podataka koje su optimizovane za analitičko procesiranje ili Online Analytical Processing (OLAP) sisteme. Takva arhitektura koristi OLTP baze podataka kao izvore podataka, koji se zatim iz njih čitaju i transformišu na odgovarajuće načine i u odgovarajuće formate koristeći ETL alate, a radi punjenja OLAP baza podataka. Projekti naprednog izveštavanja koji koriste navedenu arhitekturu mogu biti veoma kompleksni i zahtevaju pažljivu pripremu dizajna OLAP sistema i pažljivo planiranje ETL procesa. ETL procesi će neminovno između ostalog uključiti čišćenje i transformaciju podataka, koji se mogu nalaziti na većim brojem OLTP sistema i drugih izvora.

Studija objavljena u ovom radu je bazirana na merenju *online processing* performansi različitih komercijalnih SUBP, uključujući arhitekture koje podržavaju kako skladišta sa rednim upisom podataka, tako i skladišta sa upisom podataka po kolonama, dizajnirana za rad sa velikim količinama podataka.

Eksperimenti su izvedeni na nekoliko dostupnih SUBP sistema i to HP Vertica (Community Edition) koja ne zahteva licencu i nema vremenskih ograničenja, Microsoft SQL Server sa Microsoft Imagine licencom. Oracle sa Developer Licencom i na dva *Open Source* SUBP, MySQL sa InfoBright Engine i MariaDB ColumnStore.

4.1. Platforme baza podataka korišćene u eksperimentima

Da bi se dobili fer rezultati performansi izabranih baza podataka i to: HP Vertica, Microsoft SQL Server, Oracle, MariaDB ColumnStore i MySQL, svaka od njih je instalirana na zasebnoj virtualnoj mašini (VM), koja se pokretala koristeći VMware Workstation Player. Svaka od virtualnih mašina je podešena tako da koristi isti nivo hardverskih resursa kao što su RAM memorija i broj procesora. Na nivou podešavanja BIOS-a, na sva tri fizička računara koja su korišćena za izvođenje eksperimenata je omogućena virtualizacija. Virtualne mašine su smeštene na internim hard diskovima na svakom računaru.

Windows 10 Professional sa 16GB RAM-a je korišćen za host operativni sistem za sve virtualne mašine. Eksperimenti su izvedeni na dva tipa softverskih okruženja:

- Starije softversko okruženje koje se sastojalo od sledećih verzija operativnih sistema i baza podataka i to:
 - HP Vertica Analytic Database Community Edition 6.0.0 *relational database management system* (RDBMS) na CentOS 5.5 x86 64 operativnom sistemu (OS), (virtualna mašina je preuzeta sa adrese my.vertica.com);
 - MySQL 5.6 RDBMS sa dve odvojene instance, jedna sa InnoDB i druga sa InfoBright *storage engine* na CentOS 6.3 x86 64 Operativnom sistemu;
 - Oracle 11g Release 2 *Relational Database Management System* (RDBMS) sa Oracle *Data Warehouse* šablonom baze, na CentOS 6.5 x86 64 Operativnom sistemu;
 - Microsoft SQL Server 2014 Enterprise edition RDBMS na Microsoft Windows Server 2008 SP1 x64 Operativnom sistemu.
- Novije softversko okruženje koje se sastojalo od sledećih verzija operativnih sistema i baza podataka i to:
 - HP Vertica Analytic Database 9.0.1 Community edition preinstalirana na Centos 7 x86 64 operativnom sistemu (virtualna mašina je preuzeta sa adrese my.vertica.com);
 - MariaDB ColumnStore baza podataka verzije 10.2.13 instalirana na Centos 7 x86 64 operativnom sistemu;
 - Oracle DB Server 12c, edicija 12.1.0 Enterprise instalirana na Centos 7 x86 64 operativnom sistemu;
 - Microsoft SQL 2017 Server (verzije 14.0) Standard edition instaliran na MS Windows Server 2016 operativnom sistemu.

Za potrebe izvođenja eksperimenata su korišćeni sledeći računari:

1. Toshiba Qosmio X775 sa I7-2630QM procesorom i 2 GHz *clock* brzinom i 16GB RAM,
2. HP Envy 17-r104nn sa I7 6700HQ procesorom i 2.6 GHz *clock* brzinom i 16GB RAM,
3. Ne-brendirani desktop računar sa I7-2600K procesorom i 3.4 GHz *clock* brzinom i 16GB RAM.

Virtuelne mašine postavljene na svakom fizičkom računaru su tokom eksperimenata konfigurisane da koriste 4GB i 8GB RAM memorije.

Pre izvršenja eksperimenata svi sistemi za upravljanje bazama podataka su samo instalirani bez dodatnog konfiguirisanja. To je za posledicu imalo da se brzine izvršavanja razlikuju i za red veličine. Zato se pristupilo proučavanju podešavanja parametara svakog SUBP-a ponaosob (veličina memorije potrebne za sortirajuće bafere, veličina memorije potrebne za merge-join bafere itd.), pa su nakon podešavanja brzine izvršenja bile ujednačenije. U nastavku je za svaki SUBP opisan način podešavanja tih konfiguracija.

Prvi test koji je izvršen na svakom fizičkom računaru je provera performansi svakog od njih. Za taj test je korišćen UserBenchMark alat (<https://www.userbenchmark.com>) koji je izvršen na svakom računaru. Rezultati testa su dati u tabeli 1.

Test \ Processor	I7-2630QM 2.0 GHz	I7-6700HQ 2.6 GHz	I7-2600K 3.4 GHz
CPU benchmark (total throughput)			
4Core Integer throughput	240.3	264.0	315.4
4Core Float throughput	222.0	285.5	322.6
4Core Mixed throughput	234.4	262.7	341.3
RAM benchmark (memory latency)			
32MB mem. latency (ns)	72.13	74.77	59.38
64MB mem. latency (ns)	73.83	74.77	60.05
128MB mem. latency (ns)	75.56	77.35	61.41
HDD benchmark (HDD Read)			
Sequent. Read (MB/s)	76.7	65.3	143.6
4k Read (MB/s)	0.8	2.0	0.7

Tabela 1. Benčmark test parametri za računare korišćene u eksperimentima

Na nekim bazama podataka instaliranim u starijem okruženju kao što su: MySQL 5.6 sa InnoDB, Oracle 11g R2 i MS SQL Server 2014 (DataBase 1 koja koristi RowStore princip skladištenja podataka) kreirani su standardni indeksi nad spoljnim ključevima u *fact* tabelama.

U HP Vertica 6.0 bazi su kreirana ograničenja (*constraints*) nad spoljnim ključevima u *fact* tabelama. Navedene aktivnosti obezbeđuju projekcije odgovarajućih veza u toku izvršavanja upita, što će za rezultat imati značajno bolje performanse. Na MS SQL Serveru 2014 (DataBase 2 koja koristi ColumnStore princip skladištenja podataka) kreirani su ne-klasterovani indeksi nad kompletnim tabelama, dok kod MySQL baze podataka koja koristi InfoBright *engine*, indeksi nisu kreirani. U novijem okruženju nad Oracle 12c, HP Vertica 9.0 i MariaDB, ColumnStore indeksi nisu kreirani. Na MS SQL Server 2017, kreirani su ne-klasterovani indeksi nad svim tabelama.

Od dodatnih podešavanja treba napomenuti da su konfiguracioni fajlovi *my.cnf* u MySQL, *my-ib.cnf* u MariaDB i *sp-file* u Oracle podešeni na vrednost RAM memorije koja je dodeljena virtualnoj mašini. Takođe, kod Oracle baze, za opciju od 4GB RAM memorije dodeljene virtualnoj mašini, podešeni su dodatni parametri: *sga_target* i *pga_aggregate_target* na 1600MB, *db_cache_size* na 312MB, *memory_target* na 3136MB, *sga_max_size* na 1920MB i *inmemory_size* (za *In-memory*) na 768MB. Za opciju od 8GB RAM memorije dodeljene virtualnoj mašini, svi prethodno navedeni parametri su duplirani. Slična podešavanja su postavljena u fajlovima *my.cnf* i *my-ib.cnf* gde je 70% od ukupne količine RAM memorije dodeljene virtualnoj mašini (2867MB / 5734MB) dodeljeno *innodb_buffer_pool_size* (2867MB za 4GB i 5734MB za 8GB), *key_buffer_size* je podešen na 1024MB/2048MB i *myisam_sort_buffer_size* je postavljen na 256MB/512MB. Za MS SQL i HP Vertica servere korišćen je *automatic memory management* za dodeljivanje označene memorije virtualnoj mašini.

4.1.1. HP Vertica baza podataka

HP Vertica predstavlja analitičku bazu podataka i namenjena je za skladištenje i rad sa velikom količinom podataka i nudi odlične performanse za korišćenje kod *Data Warehouse* projekata, kao i kada je potrebno obezbediti najbolje performanse čitanja podataka. Imajući u vidu da je ova baza dizajnirana za analitičke procese, njene performanse za transakcione operacije su slabije. Besplatne verzije HP Vertica (HP Vertica Community Edition) koje su korišćene u ovoj studiji su Enterprise Edition (ver. 6.0) i Analytics Platform (ver. 9.1). One su dostupne za preuzimanje u obliku virtualnih mašina (<https://www.vertica.com/log-in>) sa preinstaliranim operativnim sistemom CentOS (64bit verzija) i HP Vertica bazom na 1-3 noda, radi analize do 1 TB podataka, odnosno do 500GB za verziju 6.0. Dozvoljeno je profesionalnim korisnicima da ažuriraju licencu radi povećanja klastera za masivno paralelno procesiranje ili MPP (*Massively Parallel Processing*), koje obezbeđuje dobre performanse za rad sa egzabajt veličinama podataka. Ova baza podataka beleži veliki broj procesiranih redova podataka u sekundi izvršavajući analitičke zahteve. Iste performanse važe ne samo kod izvršenja upita, već i kod procesa učitavanja podataka u bazu. Procesi učitavanja velike količine podataka odjednom

(*Bulk load*) moraju biti brzi i ne smeju da sprečavaju ili neopravdano usporavaju izvršavanje upita, koji se dešavaju u isto vreme kada i učitavanje podataka. Vertica koristi sistem organizacije tabela u projekcije, koje su u stvari sortirani podskupovi atributa tabele. Svaka projekcija poseduje specifičan redosled sortiranja podataka. Za potrebe skladištenja podataka mogu se koristiti dva pristupa i to: *Read Optimized Store* (ROS) i *Write Optimized Store* (WOS). ROS smešta podatke na standardni fajl sistem na disku koristeći višestruke ROS kontejnere. Svaki ROS kontejner, logički gledano sadrži neki broj kompletnih skupova podataka sortiranih po redosledu sortiranja date projekcije, skladišten kao par fajlova po koloni. Jedan fajl sadrži aktuelne podatke a drugi fajl sadrži pozicije indeksa. HP Vertica je su suštini *Column-store* baza, jer se fajlovi koji nose sadržaje kolona mogu učitavati i iz fizički različitih skladišta. Za potrebe optimizacije opterećenja pri izvršavanju postavljenih upita Vertica koristi Database Designer (DBD) alat koji automatski pronalazi odgovarajuće skupove projekcija. Takođe, treba napomenuti i da ovu bazu karakterišu i visoke performanse operacija sortiranja i kompresije.

4.1.2. Oracle 11g R2 sa Data Warehouse šablonom i 12c sa In-Memory Column-Store

Oracle 11g R2 baza podataka može da se instalira koristeći jedan od sledećih šablona: Transaction Processing, General Purpose, Data Warehouse i Custom Database uz pomoć ugrađenog alata Database Configuration Assistant (DBCA). Jedna od specifičnosti Data Warehouse šablona koji je korišćen pri izvođenju eksperimenata je podešavanje sistemskog parametra `star_transformation_enabled` na "true". Uključivanje ovog parametra je veoma važno kada se planiraju izvršavanja brojnih kompleksnih upita nad velikim količinama podataka, radi dobijanja najboljih performansi sistema. *Star transformation optimizer* je prvi put predstavljen u Oracle 8i radi procesiranja tzv. *Star* upita koji se najčešće koriste u *Data Warehouse* aplikacijama. U suštini, osnovni cilj star transformacije je efikasan pristup redukovanom setu podataka iz *fact* tabela kada je upit filtriran po predikatima *fact* ili *dimensions* tabela. Oracle 11g R2 Data Warehouse šablon omogućava *in-memory* paralelno izvršavanje procesa koje daje svoj doprinos korišćenjem *buffer* keša baze podataka, čija se vrednost može tačno podesiti preko sistemskog parametra `db_cache_size`. Korišćenjem paralelnog izvršavanja i pristupajući podacima u bafer kešu, podaci se mogu čitati i deset puta brže nego kada se podacima mora pristupiti sa diska. Ovaj šablon takođe omogućuje korišćenje *Bitmap* indeksa kao i *Bitmap Join* indeksa, čime se značajno dobija na performansama sistema.

Tradicionalne baze podataka smeštaju podatke na disk u redovima. Analitičke *in-memory* baze podataka smeštaju podatke po kolonama. Oracle je u verziji 12c implementirao oba navedena pristupa, koja su do tada bila dostupna kroz zasebne proizvode. *In-memory Column-store* daje mogućnost korisnicima da imaju transakcionu i analitičku bazu podataka u okviru jednog proizvoda. Za trajno skladištenje podataka se koristi *row* format na disku. Kada se zahtevaju

operacije čitanja/pisanja, podaci se učitavaju u tradicionalni *Row Store* bafer keš, dok u slučajevima zahteva samo za operacijama čitanja, podaci se učitavaju u *In-memory Column-store*. Kada je potrebno izvršiti transakciju koja uključuje upisivanje, ažuriranje ili brisanje, novi podaci će se automatski pojaviti u oba skladišta.

Administrator Oracle 12c može izabrati koje će specifične grupe kolona, kompletne tabele, pogleda ili particije tabela da čuva u *In-memory Column-store*. Iz tog razloga je ovo skladište dobar izbor za procese skeniranja podataka koji zahtevaju uslovno filtriranje, upite koji vraćaju manji broj kolona iz tabela koje poseduju veći broj kolona, upite koji spajaju manje tabele u velike tabele i upite koji agregiraju podatke. *In-memory Column-store* opcija je u potpunosti kompatibilna sa drugim svojstvima Oracle baza podataka, kao što su kompresija, enkripcija i particionisanje tabela. Količina memorije koja se dodeljuje za *In-memory Column-store* se podešava koristeći sistemski parametar `inmemory_size`, dok se objekti kao što su tabele, kolone i sl. ubacuju u skladište koristeći klauzulu *INMEMORY* u DDL naredbama kao što su *CREATE TABLE*, *ALTER TABLE* itd.

4.1.3. MS SQL sa ColumnStore Indeksom

MS SQL server ima mogućnost kreiranja *Column-store* indeksa i predstavlja ne-hibridnu implementaciju, zbog toga što skladišti sve podatke svake kolone na zasebnim stranicama. Rezultat ovog pristupa se ogleda u boljim performansama čitanja i efikasnijem korišćenju memorije. *Column-store* indeksi čuvaju podatke kolona u kompresionom formatu koji je dizajniran za brzo čitanje kompletnog sadržaja kolona. Skup redova koje je potrebno skladištiti se prvo deli u grupe redova, pri čemu se svaka grupa sastoji od fiksnog broja kolona. Svaka *row* grupa se se kodira i kompresuje nezavisno. Rezultat ovakvog pristupa je jedan *column* kompresovani segment za svaku uključenu kolonu. Svaki *column* segment se skladišti kao zaseban BLOB objekat (*Binary Large Object*).

Microsoft je proširio svojstva postojećeg *query processing engine* za *Data Warehouse*. Takođe, u okviru ovog rešenja se nalazi i optimajzer upita koji može da izabere najbolji tip operatora (između opcija *row-at-a-time* i *batch-at-a-time*) za date procese. Korišćenjem navedenog optimajzera se dobija veća brzina izvršenja upita.

Microsoft SQL Server *Columnstore index* je dizajniran da radi sa *Data Warehouse* implementacijama, posebno kod onih sa velikim *fact* tabelama. Odlične performanse karakterišu procese učitavanje velike količine podataka odjednom (*bulk loads*) i upitima koji analiziraju velike količine podataka. *Column-store* indeksi nude odlične performanse prilikom skeniranja kompletnih tabela, pri čemu nisu dizajnirani za pretragu specifičnih vrednosti.

Mogu se kreirati dva tipa indeksa, klasterovani i neklasterovani. Klasterovani *Column-store* indeks se ne može kombinovati sa drugim indeksima. On je primarni i jedini za celu tabelu i obuhvata sve kolone. Sa druge strane, ne-klasterovani indeks je sekundarni indeks koji može da indeksira podskup kolona u klasterovanom indeksu i zahteva dodatni skladišni prostor za kopije kolona u indeksu. Za razliku od klasterovanih indeksa, neklasterovani se mogu kombinovati sa drugim indeksima u tabeli.

4.1.4. MariaDB ColumnStore

MariaDB je distribuirana baza podataka koja koristi masivnu paralelizaciju i skladišti podatke po kolonama. Njena osnova potiče od InfiniDB 4.6.7 baze podataka, koja je primenjena na MariaDB 10.1 sistemu. Karakteristike ove baze podataka su brzo čitanje podataka iz skladišta, kao i brzo izvršavanje upita nad velikim skupom podataka u bazi. Takođe, MariaDB pruža mogućnost izvršavanja paralelnih upita, što u praksi znači da se upiti mogu izvršavati istovremeno na više nodova, čime se ubrzava dobijanje rezultata upita. Zbog kolumnarnog skladištenja podataka, pruža se mogućnost jednostavnog i brzog dodavanja ili ukidanja kolona, čak i u *online* režimu. Podržano je skaliranje za *Big Data* na nivou izvršavanja procesa obrade podataka veličine petabajta. Takođe, podržana je linearna skalabilnost, kompresija podataka kao i vertikalno particionisanje.

4.1.5. MySQL sa InnoDB i InfoBright engine

MySQL baza podataka je predstavnik tradicionalnih baza podataka koja se masivno koristi kako u poslovnim okruženjima, tako i za skladištenje podataka za internet aplikacije i servise. Ona je dizajnirana za OLTP transakcije i u osnovnoj konfiguraciji koristi InnoDB transakcioni *engine* i NDBCLUSTER *engine* za skladištenje podataka.

Na osnovu preporuka iz referentnog vodiča za MySQL, za potrebe *Data Warehouse* implementacija, InfoBright *engine* je najbolji izbor na osnovu uporedne analize dostupnih *engine*. Neke od karakteristika InfoBright *engine* su niska cena skladištenja podataka, korišćenje *knowledge grid* tehnike, brzo učitavanje podataka koristeći *high-speed data loader*, kolumnarni upis podataka i visoki nivoi kompresije podataka, radi pružanja odličnih *Data Warehouse* performansi.

4.2. Priprema test okruženja

Za potrebe izvođenja eksperimenata sve korišćene baze podataka su pripremljene na sličan način. Ista šema baza podataka kao i isti skup tabela i sadržaja je postavljen na svakoj bazi podataka. Korišćena je Vmart baza podataka koje je primer baze podataka postavljen od strane proizvođača u Vertica virtualnoj mašini. Ova ugrađena baza podataka se najčešće koristi za testiranje karakteristika i performansi Vertica *engine*. Radi definisanja šeme Vmart baze podataka se može koristiti ugrađena skripta `vmart_define_schema.sql`, koja se nalazi u folderu `/opt/vertica/examples/VMart_Schema/` Vertica virtualne mašine. Imajući u vidu da se u eksperimentima koriste različite baze podataka, ova inicijalna skripta je posebno prilagođena za svaku bazu podataka. Vmart baza podataka se sastoji od tri šeme: Public, Store i Online_Sales. Za izvođenje eksperimenata četiri *fact* tabele su posebno značajne, zbog ukupnog broja redova koje sadrže i to: `inventory_fact` i `store_orders_fact` sa 300.000 redova i `store_sales_fact` i `online_sales_fact` sa 5.000.000 redova.

Postavljanje odgovarajućeg okruženja je započeto instaliranjem primera Vmart baze podataka na Vertica sistemu, izvršavanjem odgovarajuće Vmart instalacione skripte. Zatim su sve tabele izvezene u fajlove sa CSV formatom. Originalne skripte za kreiranje tabela koristeći DDL komande su posebno prilagođene koristeći sintakse drugih baza podataka, koje se koriste u eksperimentu.

Vmart baza podataka je zatim instalirana na sve SUBP izvršavajući prvo skripte za kreiranje šema, a zatim su na svakoj od njih pokrenuti odgovarajući alati za grupno učitavanje podataka iz CSV datoteka.

U Vertica 9.0 je korišćen alat Vertica Management Console i Naredba „*COPY*“ za učitavanje podataka. Sličan pristup za učitavanje podataka iz CSV datoteke je korišćen u Vertica 6.0 bazi, korišćenjem iste naredbe u RazorSQL klijentskoj aplikaciji. Kod MySQL i MariaDB je korišćena naredba „*Load data*“ iz MySQL Workbench alata. Za punjenje Oracle baze je korišćena „*Sql Loader*“ komanda iz linux terminalskog shell-a sa Oracle korisničkim akreditivima, dok je kod MS SQL Servera korišćen BCP program izvršen iz komandne linije.

Za potrebe povezivanja na odgovarajuću bazu podataka i izvršavanje upita upotrebljena su četiri različita alata postavljena na odgovarajućim virtualnim mašinama. Za Vertica 6.0 bazu je korišćen alat RazorSQL, za Oracle bazu alat Oracle SQL Developer, za MySQL i MariaDB Columnstore bazu MySQL Workbench alat, dok je SQL Server Management Studio korišćen za povezivanje sa MS SQL Serverom. Alati RazorSQL, SQL Developer i MySQL Workbench poseduju ugrađenu funkcionalnost koja ima mogućnost prikaza vremena izvršenja upita. Radi prikazivanja vremena izvršenja operacija u SQL Management Studiu, korišćena je opcija

‘*Include Client Statistics*’ koja prikazuje „ukupno vreme izvršenja“, zajedno sa drugim relevantnim statističkim parametrima povezanim sa procesima izvršenja upita.

Eksperimenti su koncipirani na izvršavanju različitih *ad/hoc* upita i merenju njihovog vremena izvršenja na svakoj od pripremljenih baza podataka. Kreirana su ukupno četiri upita koja su imala za cilj da učitaju i procesiraju sve redove iz *fact* tabela, bez korišćenja filtera (preko *WHERE* klauzule), a radi maksimalnog opterećenja sistema. Svi definisani upiti su dizajnirani koristeći PL/SQL sintaksu, pri čemu su izvedene odgovarajuća podešavanja u SQL sintaksama, zbog izvršenja na različitim sistemima.

Postavljeni cilj **Upita 1 (Q1)** je testiranje performansi agregacije u jednoj tabeli činjenica od 5 miliona redova, spojenoj *inner-join* vezom sa jednom kolonom u dimenzionoj tabeli *promotion_dimension*, koja se sastoji od 1,000 redova i *inner-join* vezama sa dve kolone u dimenzionoj tabeli *product_dimension*, koja se sastoji od 60.000 redova. Polje vrednosti valute je korišćeno za agregaciju, a grupisanje je izvedeno po dve kolone iz dimenzionih tabela. Definisani upit treba da prikaže rezultate ostvarenog profita, koji je grupisan po kategorijama proizvoda i vrstama sniženja cena.

```
SELECT prom.price_reduction_type, prod.category_description,  
sum(o.gross_profit_dollar_amount) AS sum_profit  
FROM online_sales.online_sales_fact o  
INNER JOIN public.promotion_dimension prom ON  
    o.promotion_key = prom.promotion_key  
INNER JOIN public.product_dimension prod ON  
    o.product_key = prod.product_key AND o.product_version =  
    prod.product_version  
GROUP BY prom.price_reduction_type, prod.category_description;
```

Upit 2 (Q2) je dizajniran na način da izvrši testiranje performansi agregacije u jednoj tabeli činjenica od 300.000 redova, spojenoj *inner-join* vezom sa jednom kolonom u dimenzionoj tabeli *warehouse_dimension*, koja sadrži 100 redova i *inner-join* vezama sa dve kolone u dimenzionoj tabeli *product_dimension* sa 60.000 redova. Polje količine je korišćeno za agregaciju, a grupisanje je izvedeno po dve kolone iz dimenzionih tabela. Definisani upit treba da prikaže broj proizvoda koji se čuvaju u skladištima, grupisan po regionima i kategorijama proizvoda.

```

SELECT w.warehouse_region, p.category_description, sum(i.qty_in_stock)
as sum_qty
FROM public.inventory_fact i
INNER JOIN public.warehouse_dimension w ON
    i.warehouse_key = w.warehouse_key
INNER JOIN public.product_dimension p ON
    i.product_key = p.product_key AND i.product_version =
p.product_version
GROUP BY w.warehouse_region, p.category_description;

```

Upit 3 (Q3) je pripremljen za testiranje performanse agregacije u jednoj tabeli činjenica sa 5 miliona redova, spojenoj *inner-join* vezama sa tri dimenzione tabele i to: datum_dimension sa 1.826 redova, store_dimension sa 250 redova i customer_dimension sa 50.000 redova. Ovaj upit sadrži dve funkcije agregacije sa jednom *CASE* klauzulom. Pored toga, korišćena je i *ROUND* funkcija u okviru funkcije agregacije. Kreirane su veze (*joins*) po jednoj koloni nad sve tri dimenzione tabele. Definisani upit treba da prikaže profite lanca supermarketa Vmart i procenat ostvarenog profita po regionima kupaca, grupisane po kalendarskim godinama

```

SELECT d.calendar_year, ROUND((SUM(CASE WHEN
cd.customer_region = sd.store_region THEN 1 ELSE 0 END) / COUNT(*)) *
100, 2) AS percentage_same_region,
SUM(gross_profit_dollar_amount) AS sum_profit
FROM store.store_sales_fact sf
INNER JOIN public.date_dimension d ON
    sf.date_key = d.date_key
INNER JOIN store.store_dimension sd ON
    sf.store_key = sd.store_key
INNER JOIN public.customer_dimension cd ON
    sf.customer_key = cd.customer_key
GROUP BY d.calendar_year;

```

Upit 4 (Q4) je dizajniran za testiranje performansi agregacije u jednoj tabeli činjenica sa 300.000 redova bez postavljanje bilo kakvih relacija sa tabelama dimenzija, ali sa višestrukim datumskim i matematičkim funkcijama kao što su: *EXTRACT*, *DATEDIFF*, *ROUND*, *SUM*, *AVG*. Ovaj upit je pripremljen sa ciljem merenja brzine procesa *online* agregacije neagregiranih podataka. Definisani upit treba da prikaže prosečno vreme u danima, koje je potrebno transportnim firmama da isporuče robu Vmart magacinu, kao i procenat kašnjenja isporuke. Izveštaj treba da bude grupisan prema imenima pošiljaoca i kalendarskim godinama izvršene isporuke.

```
SELECT shipper_name, EXTRACT(year from date_ordered) AS  
calendar_year,  
ROUND((SUM(CASE WHEN date_delivered > expected_delivery_date  
THEN 1 ELSE 0 END)/COUNT(*))*100, 2) AS percentage_latency,  
ROUND(AVG(DATEDIFF(date_delivered, expected_delivery_date)), 2) AS  
delivering_days,  
ROUND(AVG(DATEDIFF(date_ordered, date_shipped)), 2) AS  
shipping_days  
FROM store.store_orders_fact  
GROUP BY shipper_name, calendar_year;
```

4.3. Rezultati istraživanja

Posebna pažnja je posvećena pripremi okruženja i izvršavanju testova da bi se dobili fer rezultati. Prilikom preliminarnog testiranja uočeno je da postoji uticaj keširanih podataka bez obzira da li se u periodu između dva ista testa restartuju servisi baze podataka. Takođe, uočeno je da se ponovljeni upit pokrenut posle restarta servisa baza podataka brže izvršava u odnosu na izvršavanje istog upita posle restartovanja virtualne mašine. Iz tog razloga, prilikom vršenja eksperimenata svaki upit je izvršavan nakon ponovnog pokretanja virtualne mašine. Takođe, kod preliminarnog testiranja je prilikom pokretanja upita na MS SQL serveru uočeno da je vreme izvršenja ponovljenog upita bilo brže u odnosu na prvo, bez obzira što je virtualna mašina u međuvremenu restartovana. Razlog ovakvog ponašanja je u tome što je MS SQL nakon izvršenog prvog upita sačuvao statističke objekte u tabelama činjenica i dimenzija, te koristio te podatke prilikom sledećeg izvršenja upita. Da bi se dobili fer rezultati merenja, nakon svakog izvršenog upita su generisani statistički podaci obrisani.

Upiti su vršeni nakon početnog smanjenja opterećenja sistema (ispod 0,2) nakon pokretanja virtualne mašine. Princip merenja rezultata je postavljen na način da se svaki test/upit izvršavao tri puta (poštujući prethodno navedeno pravilo izvršenja jednog upita po pokretanju virtualne mašine), a zatim je za izmerenu vrednost uziman prosečan rezultat sva tri merenja. U okviru prvog eksperimenta se nalaze upiti pokrenuti na „starijim“ OS i SUBP, a u okviru drugog eksperimenta upiti pokrenuti na „novijim“ OS i SUBP.

Prilikom vršenja eksperimenata, utvrđeno je da su brzine izvršavanja upita Q1 i Q3 na MySQL-a sa InnoDB *engine* i MS SQL Server-u sa standardnim *RowStore* indeksima na najsporijem računaru bile veoma male, tako da se upiti/eksperimenti na ovim bazama podataka u drugim hardverskim konfiguracijama nisu pokretali. Dobijeni rezultati brzina izvršenih upita po eksperimentima su prikazani u tabelama 2 i 3.

Tabele sa rezultatima merenja brzine upita su formatirane za prikazivanje više dimenzija i to:

- Arhitektura računara prikazana kroz performanse procesora i RAM memorije,
- Količina RAM memorije koja je dodeljena svakoj virtualnoj mašini,
- Korišćeni sistem baza podataka i
- Broj/oznaka upita (Q1-Q4).

Processor / RAM	Queries	MS SQL 2008 (columnstore)		HP Vertica 6.0		MySQL 5.6 (IB)		Oracle 11g datawarehouse		MS SQL (rowstore)	MySQL (InnoDB)
		4GB	8GB	4GB	8GB	4GB	8GB	4GB	8GB	4GB	4GB
I7-2630QM 2.0GHz DDR3 RAM 1333MHz CL9	Q1	3182	3088	4423	2750	6023	5018	9531	8026	79633	214399
	Q2	2090	2037	456	272	1018	1015	1699	1056	710	6433
	Q3	3416	3244	3899	3329	6012 ⁻	6019 ⁺	9526	9522	12320	87719
	Q4	1014	858	318	318	1010 ⁻	1012 ⁺	1009 ⁻	1012 ⁺	103	1321
I7- 6700HQ 2.6GHz DDR3 RAM 1600MHz CL11	Q1	2303 ⁻	2414 ⁺	2873	1991	5151	4527	7037	6520		
	Q2	2086	2187	550	432	508	508	507 ⁻	509 ⁺		
	Q3	2144	2109	2210 [*]	1575	5013 ⁻	5030 ⁺	8022	7600		
	Q4	952	1123	303	280	1008	1007	1009 ⁻	1011 ⁺		
I7-2600K 3.4GHz DDR3 RAM 1600MHz CL11	Q1	1981	1775	2463	1539	4017 ⁻	4021 ⁺	5016	5015		
	Q2	998	951	415	376	571	543	506	506		
	Q3	1887	1703	1579	1075	4666	4513	5521	5514		
	Q4	530	530	203 ⁻	207 ⁺	1007	507	1011	1011		

Tabela 2. Vreme (u ms) potrebno za izvršavanje upita za različite „starije“ SUBP na različitim računarima i različitoj memoriji dodeljenoj u Eksperimentu 1.

Processor / Clock speed	Queries	MS SQL 2017 (columnstore)		HP Vertica 9.0.1		MariaDB columnstore 10.2.13		Oracle 12c In- Memory Column Store	
		4GB	8GB	4GB	8GB	4GB	8GB	4GB	8GB
I7-2630QM 2.0GHz DDR3 RAM 1333MHz CL9	Q1	3358	2414	3490	1720	3936	3779	3272	2915
	Q2	844	836	357	323	1305	1262	498	382
	Q3	2594	2875	2075	1059	4452	4282	4312	4148
	Q4	875	795	315	311	1042	1022	922	610
I7- 6700HQ 2.6GHz DDR3 RAM 1600MHz CL11	Q1	1575	1515	2769	1483	3124	2927	2289	2193
	Q2	907	440	413	282	542	489	473	252
	Q3	2008	1915	2265 [*]	936	3889	3285	3844	3671
	Q4	718 ⁻	796 ⁺	200 ⁻	222 ⁺	732	716	646	584
I7-2600K 3.4GHz DDR3 RAM 1600MHz CL11	Q1	1516	1500	2498	1371	2741 ⁻	2745 ⁺	2057	1928
	Q2	392	461	236	214	466 ⁻	483 ⁺	253	222
	Q3	1606	1579	1482	803	2347	2308	2930	2778
	Q4	516	497	292	163	344 ⁻	404 ⁺	433	363

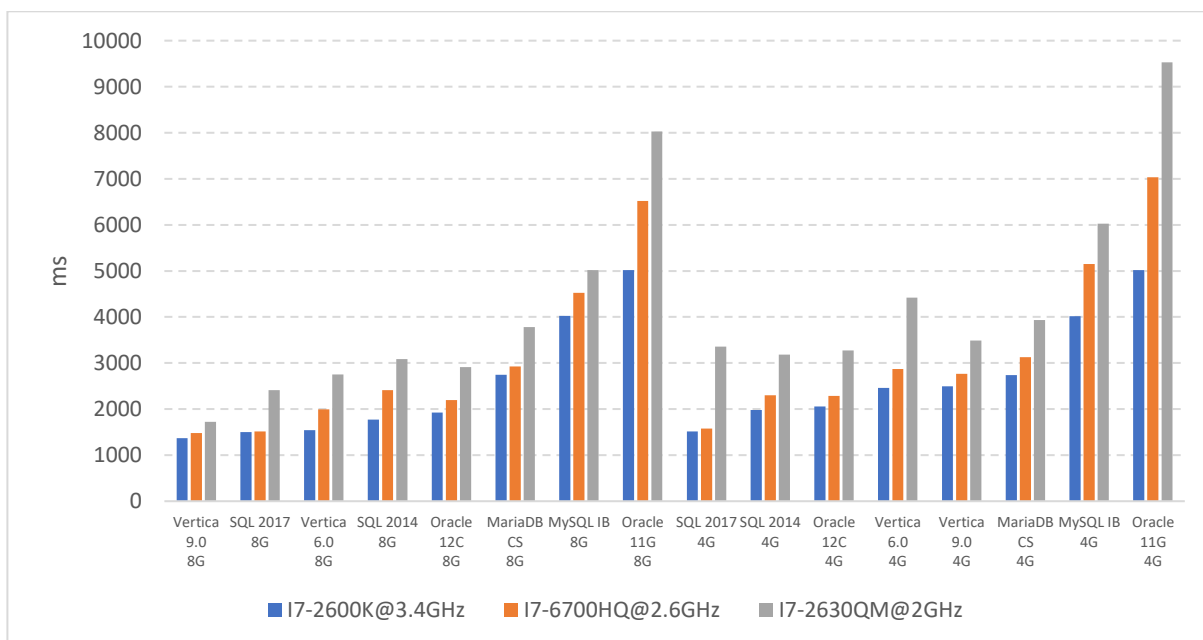
Tabela 3. Vreme (u ms) potrebno za izvršavanje upita za različite „novije“ SUBP na različitim računarima i različitoj memoriji dodeljenoj u eksperimentu 2

Radi prikazivanja izuzetaka u rezultatima u obe tabele su korišćeni simboli. Simboli „+“ i „-“ u superskriptu označavaju „izuzetke“, da se isti upit izvršava brže koristeći manje memorije na

istom računar. Simbol „*“ u superskriptu takođe označava drugu vrstu „izuzetka“, da se isti upit brže izvršava na „starijoj“ verziji SUBP na istom računar.

Izmereni rezultati su opravdali očekivanja. U većini slučajeva upiti su se izvršavali brže na boljim računarima, koristeći više RAM-a i na „novijim“ verzijama SUBP. Vreme izvršenja upita Q1 i Q3 nad tabelama činjenica sa 5M redova je bilo najbrže na Vertica 9.0 SUBP koristeći 8G RAM-a na svim računarima. U opciji sa virtualnim mašinama kod kojih je dodeljeno 4GB memorije, upit Q1 se brže izvršavao na MS SQL 2017, a upit Q3 je bio brži na Vertica 9.0. Navedeni upiti su se najsporije izvršavali u Oracle 11g, što je i očekivano, jer on nema implementirane *ColumnStore* indekse. Može se zaključiti da bez obzira na najsporije izvršenje upita u poređenju sa sistemima koje podržavaju *ColumnStore* indekse, brzina izvršenja upita u Oracle 11G je značajno bolja u poređenju sa vremenom izvršavanja istih upita kod MS SQL 2014 i MySQL (InnoDB) baza podataka koje koriste *RowStore* pristup.

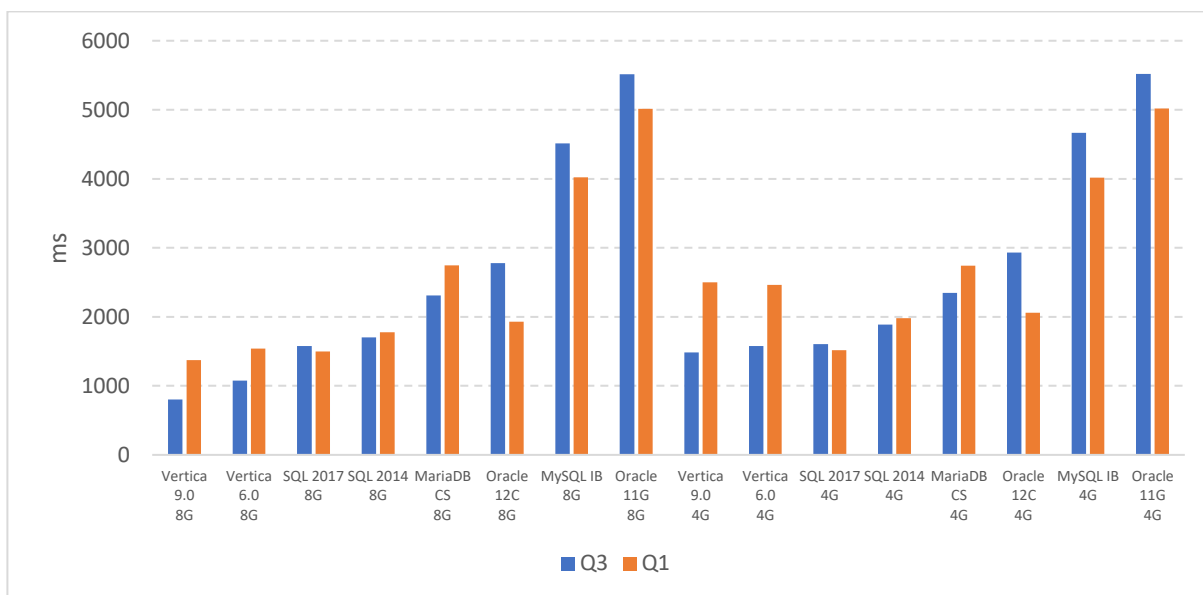
Poređenje vremena izvršavanja upita Q1 u odnosu na arhitekture sistema, količine RAM-a dodeljene virtualnim mašinama i sisteme baza podataka prikazano je na grafikonu 1. Jasno se može uočiti da je vreme izvršenja upita u skladu sa performansama arhitekture računara i da se upit Q1 uglavnom brže izvršava na virtualnim mašinama sa većom dodeljenom RAM memorijom. Redosled prikaza sistema baza podataka je sortiran sa leva na desno, shodno brzini izvršenja upita na svakom računar i podeljen u dve grupe na osnovu dodeljene količine memorije.



Grafikon 1. Brzina izvršavanja upita Q1

Na osnovu prikazanih rezultata u obe tabele se ne može lako videti koji od upita Q1 i Q3 su bili brže izvršeni na različitim platformama. Upit Q1 se izvršavao brže od upita Q3 u MS SQL 2017 u svim slučajevima (osim na računaru sa I7-2630QM procesorom i 4GB RAM memorije dodeljene virtualnoj mašini), u Oracle 12C u svim slučajevima, u Oracle 11GR2 u svim slučajevima (osim na najsporijem računaru sa 4GB RAM memorije dodeljene virtualnoj mašini), u MariaDB CS na sporijim računarima, u MySQL IB na najbržem računaru, kao i na sporijim računarima sa 8GB RAM memorije dodeljene virtualnoj mašini. Upit Q3 se izvršavao brže u drugim slučajevima. Takođe se može uočiti, da je slična situacija je i sa upitima Q2 i Q4. Kod nekih slučaja upit Q2 se izvršavao brže (u Oracle 12c na svim računarima, na bržim računarima u MySQL IB i Oracle 11G, itd.), dok je kod nekih drugih upit Q4 bio brži. Upiti Q2 i Q4 su bili izvršeni sa približnim brzinama na različitim platformama baza podataka bez obzira na tip indeksa, zbog relativno malog broja redova u tabeli činjenica („samo“ 300.000 redova u tabeli).

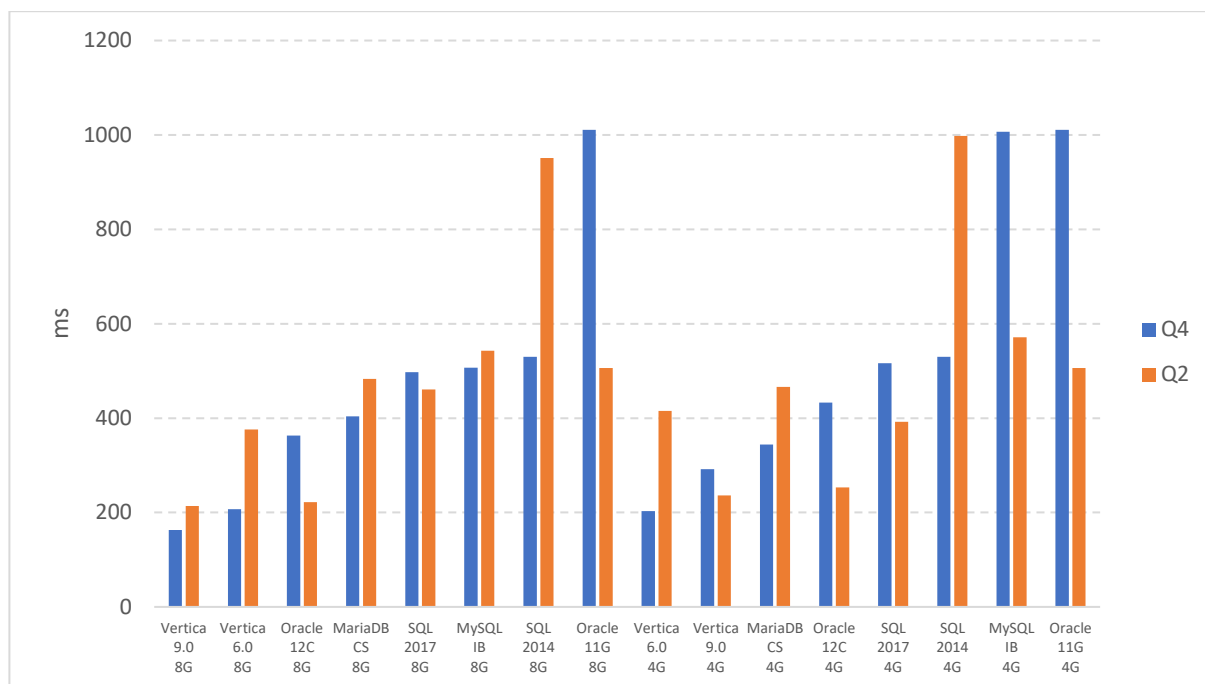
Pošto će se u produkcionim kruženjima koristiti snažniji računari, reprezentativni primer koji se može izabrati je primer poređenja vremena izvršavanja upita Q1 i Q3 sa *fact* tabelom od 5M redova na najbržem računaru. Rezultati ovog primera su prikazani na grafikonu 2.



Grafikon 2. Brzina izvršavanja upita Q1 i Q3 sa većim Fact tabelama na najbržem računaru koji se koristi u eksperimentima

Na grafikonu 2 su platforme baza podataka sortirane sa leva na desno prema vremenu izvršenja upita Q3. Može se videti da je vreme izvršenja upita Q1 i Q3 manje od 5,6 sekundi na svim platformama baza podataka na najbržem računaru. Najbolja vremena izvršenja su izmerena baze Vertica i MS SQL.

Analiza performansi pri izvršenju upita Q2 i Q4 se može izvršiti na osnovu prikazanih rezultata na grafikonu 3.



Grafikon 3. Brzina izvršavanja upita Q2 i Q4 sa manjim Fact tabelama na najbržem računaru koji se koristi u eksperimentima

Upiti Q2 i Q4 su se najbrže izvršavali na Vertica platformi, a zatim slede Oracle 12C i MariaDB ColumnStore. Najbolje vreme izvršavanja za ova dva upita u Vertica bazi podataka je iznosilo oko 200 ms.

Kao što se i očekivalo, generalno gledano, upiti na tabelama sa većim brojem redova su se izvršavali sporije od onih sa manjim brojem redova. Sa korišćenjem snažnijih računara dobijena su bolja vremena izvršavanja upita. Međutim, sa povećanjem RAM memorije dodeljene virtualnim mašinama, nisu uvek dobijene veće brzine izvršavanja upita. Platforme baza podataka koje koriste *ColumnStore* indekse nad tabelama, kao što se i očekivalo, izvršavaju upite znatno brže nego one sa standardnim *RowStore* indeksima.

4.4. Zaključak

U okviru ove studije je prikazano poređenje performansi izvršenja upita nad velikim skupom neagregiranih podataka koristeći različite platforme baza podataka. Procesi izvršenja agregacionih upita izvršeni su u *on-line* režimu. Zahvaljujući podešavanju konfiguracija SUBP na administratorskom nivou, dobijena vremena izvršenja upita su za pojedine SUBP bila znatno kraća u odnosu na inicijalno instalirane baze bez podešavanja konfiguracionih fajlova. Na osnovu izvršenih eksperimenata dokazana je opšta pretpostavka da će kolumnarne platforme nadmašiti standardne baze podataka koje upisuju podatke u skladište po redovima. Sve izabrane platforme baza podataka nad kojima su vršeni eksperimenti: MS SQL Server, HP Vertica, Oracle, MariaDB ColumnStore i MySQL su postavljene na virtualnim mašinama koristeći VMware virtualizaciono okruženje na nekoliko računara sa različitim hardverskim performansama i sa 4GB i 8GB RAM memorije dodeljene virtuelnim mašinama. Test baza podataka koja se sastojala od četiri tabele činjenica i nekoliko dimenzionih tabela je instalirana na svakoj izabranoj platformi baza podataka. Svaka baza podataka je testirana su četiri različita upita nad tabelama činjenica koje su sadržale po 5,000,000 i 300,000 redova.

Najveća brzina izvršenja postignuta je očekivano na najbržem računaru korišćenom u eksperimentu, na komercijalnim SUBP: HP Vertica i MS SQL Server, kod kojih su tabele u bazi podataka indeksirane u ColumnStore formatu. Rezultati eksperimenata su pokazali da je izvršenje upita (Q1 i Q3) nad većim tabelama činjenica (sa 5,000,000 redova) bilo brže u Vertica bazi kada je 8GB RAM memorije dodeljeno virtualnoj mašini, dok je upit Q1 brže izvršen na MS SQL Serveru sa 4GB RAM memorije dodeljene virtualnoj mašini. HP Vertica je pokazala brže izvršavanje upita (Q2 i Q4) sa manjim tabelama činjenica, bez obzira na količinu RAM memorije dodeljene virtualnim mašinama. Nekomercijalne baze podataka MariaDB ColumnStore i MySQL DB sa InfoBright mehanizmom za skladištenje koje koriste ColumnStore indeksiranje su pokazale prilično dobre performanse. Oracle 12c je upite Q1 i Q2 brže izvršavao nego MariaDB CS, ali su upiti Q3 i Q4 brže izvršavani na MariaDB CS. Generalno posmatrano, svi upiti su izvršavani na svim računarima za manje od 10 sekundi (osim u slučajevima kada je baza podataka koristila indekse *rowstore* u MS SQL i MySQL). Na osnovu izmerenih rezultata preporuka je da se kolumnarne platforme baza podataka koriste za analitičke potrebe.

Na izbor odgovarajućeg rešenja za poslovanje može uticati i cena samog rešenja. U trenutku pisanja ove disertacije, na osnovu raspoloživih informacija na web sajtovima proizvođača čiji proizvodi su korišćeni u eksperimentima, a koji su podložni licenciranju je data u sledećim tačkama:

- Microsoft SQL Server se može licencirati po Per Core i Server + CAL (Client Access License) modelima, koristeći različite programe licenciranja. Cena licence SQL Server Enterprise edicije koja pokriva 2 procesorska jezgra kroz program količinskog licenciranja iznosi 15,123 USD.
- HP Vertica Community Edition je besplatna. Ukoliko je potrebno koristiti komercijalnu verziju, HP Vertica se licencira po nodu koji može imati do 16 jezgara. Vertica Premium Licenca sa uključenom 24x7 jednogodišnjom podrškom iznosi 13,930 USD
- Oracle Baza podataka se licencira po procesoru. Procesorska Licenca Enterprise Edicije iznosi 47,500 USD a na cenu licence treba dodati i Software Update i podršku koja iznosi 10,450 USD.
- MariaDB Community Server je besplatan, dok je MariaDB Enterprise podlozan licenciranju i cena zavisi od regiona korišćenja, te se za konkretne slučajeve treba obratiti proizvođaču.
- MySQL je *Open Source/besplatan* proizvod. MySQL Enterprise proizvod je podlozan licenciranju a početna cena MySQL Enterprise Edition Subscription (On premise) licence koja pokriva server sa 1-4 soketa iznosi 5,350 USD.

Imajući u vidu visok nivo integracije sa ostalim sistemima kao i odlične performanse besplatne edicije HP Vertica (HP Vertica Community Edition), poslovne organizacije bi trebalo da razmotre njeno korišćenje za potrebe izveštavanja. Ukoliko ograničenje besplatne edicija ne utiču negativno na performanse rešenja, ovaj SUBP može da predstavlja odličan izbor za mala i srednja preduzeća.

.

5. Poređenje brzina izvršenja ETL upita pri korišćenju SQL koda, SSIS ugrađenog ETL alata i Pentaho besplatnog ETL alata

Kvalitetni podaci na osnovu kojih se generišu adekvatni izveštaji i obezbeđuju odgovarajuće analitičke funkcije predstavljaju prvi preduslov za donošenje kvalitetnih poslovnih odluka. Podaci se vrlo često nalaze u brojnim izvorima koji mogu biti struktuirani, polustruktuirani i nestruktuirani, smešteni u bazama podataka kao i fajlovima različitih formata. Projekti naprednog poslovnog izveštavanja su jedni od najkompleksnijih i zahtevaju adekvatna znanja i tehnike, počevši od čišćenja i transformacije izvornih podataka, preko dizajna analitičke baze podataka (OLAP) u koju se smeštaju pripremljeni podaci, pa do BI alata zaduženih za dizajn izveštaja i prikazivanje rezultata upita donosiocima poslovnih odluka.

Najveći deo projekata naprednog izveštavanja odlazi na pripremu podataka za njihovo učitavanje u analitičku bazu podataka. Za pripremu podataka su zaduženi ETL procesi (*Extraction-Transformation-Loading*). ETL procesi započinju prvim korakom, a to je prepoznavanje odgovarajućih podataka iz izvora (najčešće iz OLTP baze podataka) i njihova ekstrakcija. Zatim se nad ekstrahovanim podacima vrše odgovarajuće transformacije podataka koje uključuju čišćenje podataka, uklanjanje pogrešnih zapisa, spajanje podataka iz različitih tabela, izvršenje modifikacija nad podacima, kako bi se na kraju podaci na brz i efikasan način mogli učitati u OLAP bazu. Za izvršenje ETL procesa se koriste ETL alati koji su u stvari softveri odgovorni za ekstrakciju podataka iz različitih izvora, njihovo čišćenje, prilagođavanje i učitavanje u *Data Warehouse* (OLAP) bazu podataka.

Razlog korišćenja OLAP baze za izvršenje analitičkih upita su značajno bolje performanse izvršenja upita u odnosu na opciju izvršavanja upita nad OLTP bazom koja je dizajnirana za brzo upisivanje podataka. Pored toga, odgovarajućom transformacijom podataka u ETL procesu radi njihovog učitavanja u OLAP bazu, smanjuje se potreba za brojem veza koje inače postoje u OLTP bazi podataka, čime se dodatno dobija na brzini izvršenja upita. Kada se podaci smeste u OLAP bazu, IT eksperti mogu da koriste robusne agregacione upite koristeći SQL ili MDX programske jezike, radi dobijanja potrebnih izveštaja. Jednostavniji način za pripremu upita je korišćenjem *Business Intelligence* (BI) alata koji su u stvari softveri koje najčešće koriste poslovni analitičari, bez potrebe poznavanja kompleksnih programskih jezika. BI se može definisati kao proces koji omogućava poslovanju da detektuje i označi najpotrebnije informacije iz ogromne količine raspoloživih podataka, radi donošenja adekvatnih poslovnih odluka.

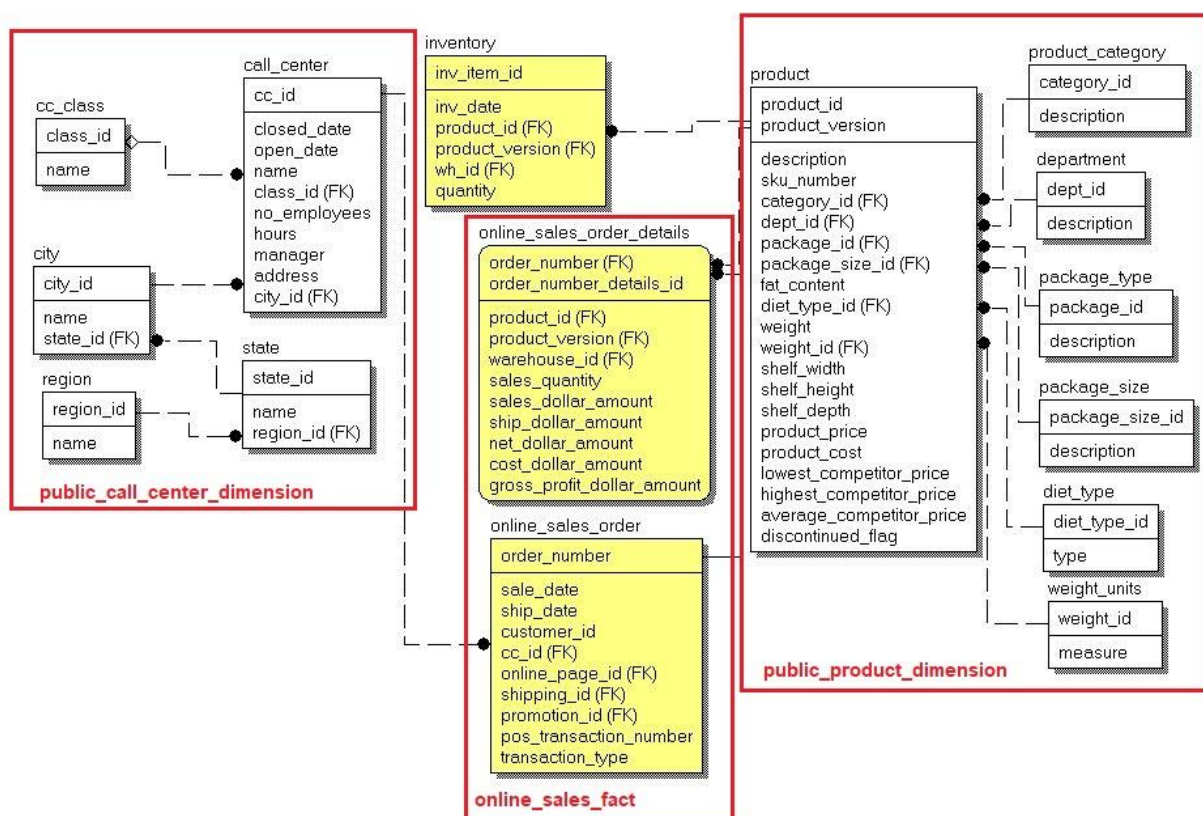
Što je količina podataka na njihovim izvorima veća to su i ETL procesi duži, te je veoma čest primer da se oni izvršavaju van radnog vremena. To je razlog zašto podaci u OLAP bazi mogu biti stariji i više od jednog dana u odnosu na one u OLTP bazi. Imajući u vidu da poslovanje nema vremena za kašnjenje i da poslovne odluke treba donositi brzo i pravovremeno, veoma je značajno da OLTP i OLAP sistemi budu frekventno sinhronizovani, bez kašnjenja.

Postavljeni cilj ove studije je merenje vremena ETL procesa radi punjenja OLTP baze velikom količinom podataka, koristeći savremene ETL alate. Imajući u vidu da brzina izvršavanja ETL procesa zavisi od performansi računara na kojima se oni izvršavaju, eksperimenti su vršeni na nekoliko računara različitih hardverskih karakteristika.

Eksperimenti su izvedeni na Microsoft SQL Server SUPB sa Microsoft Imagine licencom.

5.1. Okruženje za izvođenje eksperimenata

Za potrebe izvođenja eksperimenata kreirana je Vmart OLTP baza podataka iz postojeće Vmart OLAP baze podataka. Navedena OLTP baza se sastojala od 48 tabela od kojih su 4 tabele od 5,000,000 redova i 3 tabele od 300,000 redova bile najveće. Sve ostale tabele su imale 25,000 i manje zapisa. Imajući u vidu da je šema Vmart OLTP baze prevelika, na slici 1 je prikazan deo šeme sa 15 tabela. Neke tabele su grupisane radi transformacije kroz ETL proces.



Slika 1. Deo šeme VMart OLTP baze

MS SQL 2017 standard edicija (DreamSpark Premium | Academic Software Discounts) je korišćena kao platforma na kojoj je instalirana VMart OLTP baza podataka. Virtualna mašina baze podataka je instalirana na Windows Server 2016 operativnom sistemu korišćenjem VMware virtualizacione platforme. Virtualnoj mašini je dodeljeno 80GB prostora na disku i 8GB RAM memorije. Tabele na slici 1 koje su označene žutom tj. tamnijom pozadinskom bojom su transformisane u tabele činjenica (*fact* tabele), dok su tabele sa belom pozadinskom bojom transformisane u tabele dimenzija (*dimensions* tabele). Tabele činjenica su sadržale događaje, tj. polovne procese u obliku sledećih polja: *purchase*, *procurement*, *lending*,

payments itd. Tabele dimenzija su sadržale attribute povezane sa mestima, vremenima i drugim tabelama kao što su: *products*, *vendors*, *shippers* itd.

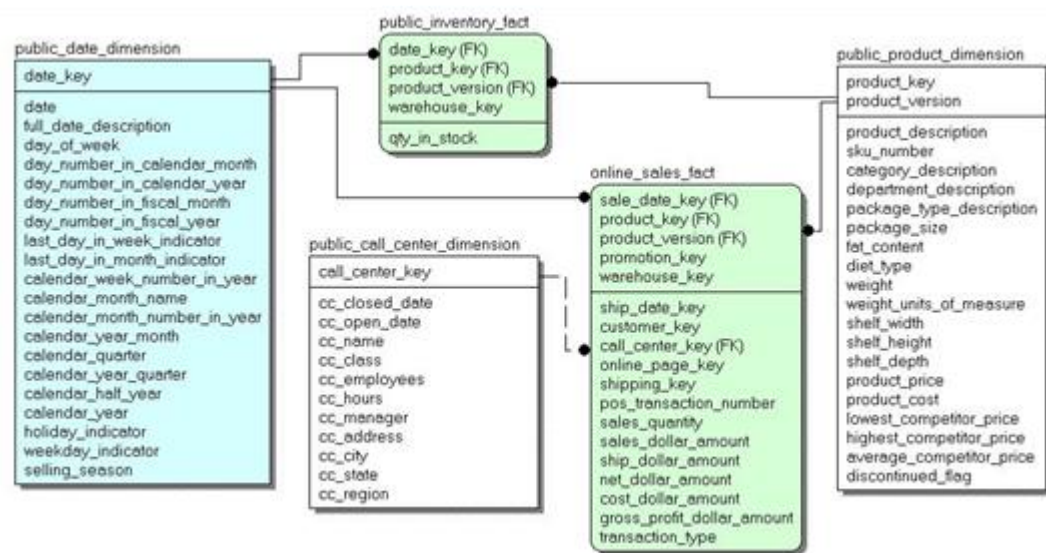
Kreirana je i baza podataka VMartD kao destinaciona OLAP baza, sa odgovarajućim tabelama činjenica i dimenzija. U okruženju za vršenje eksperimenata, OLTP Vmart baza podataka je korišćena kao izvorna baza iz kojih su se podaci ekstrahovali i transformisali radi punjenja VMartD OLAP baze podataka korišćenjem Microsoft ETL aplikacije - MS SQL Server Integration Services (SSIS) i Pentaho ETL alata.

Prvi eksperiment je zasnovan na izvršavanju ETL procesa korišćenjem Rešenja 1, kod koga su procesi kreirani korišćenjem skripti sa SQL kodom. Drugi eksperiment se sastojao od izvršavanja ETL procesa upotrebom Rešenja 2, koje je zasnovano na korišćenju raspoloživih SSIS ETL alata. Treći eksperiment je podrazumevao izvršenje ETL procesa koristeći Rešenje 3 bazirano na besplatnom Pentaho alatu (*community edition* firme Hitachi Vantara). Za svaki od navedenih eksperimenata je vršeno merenje brzine ETL procesa.

Podaci iz pet tabela OLTP baze, kao što je prikazano na Slici 1 (grupacija prikazana na levoj strani), su grupisani koristeći relacije sa tabelom ***call_center*** (***cc_class***, ***city***, ***state***, ***region***) i spojeni (transformisani) u jednu dimenzionu tabelu ***public_call_center_dimension*** u OLAP bazi podataka. Podaci iz šest tabela OLTP baze prikazane na Slici 1 (grupacija prikazana na desnoj strani) povezani relacijama sa tabelom ***product*** (***product_category***, ***department***, ***package_type***, ***package_size***, ***diet_type***, ***weight_units***) su spojeni u jednu dimenzionu tabelu ***public_product_dimension*** u OLAP bazi podataka. Podaci dve OLTP tabele ***online_sales_order*** i ***online_sales_order_details*** će biti transformisane u OLAP tabelu činjenica ***online_sales_fact***. Pored toga, podaci iz četiri grupe tabela koje su slične tabelama sa žutom (tamnijom) pozadinom na Slici 1, će biti transformisane i učitane u četiri tabele činjenica.

Cilj ETL procesa se ogleda u ekstrakciji svih podataka gore navedenih tabela, izvođenjem *join* operacija i učitavanjem transformisanih podataka u odgovarajuće tabele dimenzija OLAP baze podataka. Sumarno posmatrano, podaci iz ukupno 41 tabele OLTP baze će biti preslikane u 9 OLAP dimenzionih tabela.

Destinacione OLAP tabele koje su dobijene ETL transformacijom izvornih tabela OLTP baze podataka su prikazane na slici 2.



Slika 2. VMart OLTP tabele dobijene transformacijom OLTP tabela sa Slike 1

Analizom tabela se može videti da su podaci u atributima: *cc_class*, *cc_city*, *cc_state* i *cc_region* u tabeli *public_call_center_dimension* ekstarkovani iz atributa “name” tabela: *cc_class*, *city*, *state* i *region*. Na isti način su izvedene i transformacije podataka iz atributa “description”, “type” i “measure” iz OLTP tabela u OLAP tabelu *public_product_dimension*. Takođe se može videti da su podaci skoro svih atributa iz OLTP tabela *online_sales_order* i *online_sales_order_details* spojeni u jednu OLAP tabelu činjenica *online_sales_fact*.

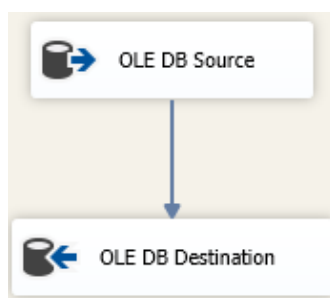
Ukoliko posmatramo samo tabele prikazane na gore navedenim slikama, jasno se može videti da su podaci 15 OLTP tabela transformisane u 5 OLAP tabela sa istom količinom podataka ali drugačijom organizacijom.

Za potrebe izvršenja eksperimenata izabrane su tri hardverske platforme sa Microsoft Windows 10 operativnim sistemom na kojima su postavljene virtualne mašine:

- Računar sa I7-11800H, 2.3GHz procesorom i SSD diskom
- Računar sa i7-9750H, 2.6GHz procesorom i SSD diskom
- Računar sa i7-2600K, 3.7 GHz procesorom i HDD diskom

5.1.1. ETL rešenje 1 – korišćenje SQL upita

SQL Server Integration Services (SSIS) konzola poseduje ugrađene komponente za kreiranje ETL poslova (jobs). Oni se mogu dizajnirati korišćenjem dve komponente, jedne za upis SQL koda za ekstrakciju i transformaciju izvornih podataka, i druge za mapiranje kolona *Select* upita nad izvornim tabelama sa kolonama u destinacionim tabelama, radi pravilnog punjenja tabela *Data Warehouse* baze. Na slici 3 su prikazane su prikazane obe komponente *OLE DB Source* i *OLE DB Destination* povezane odgovarajućom konekcijom.



Slika 3. *OLE DB Source* i *OLE DB Destination* komponente

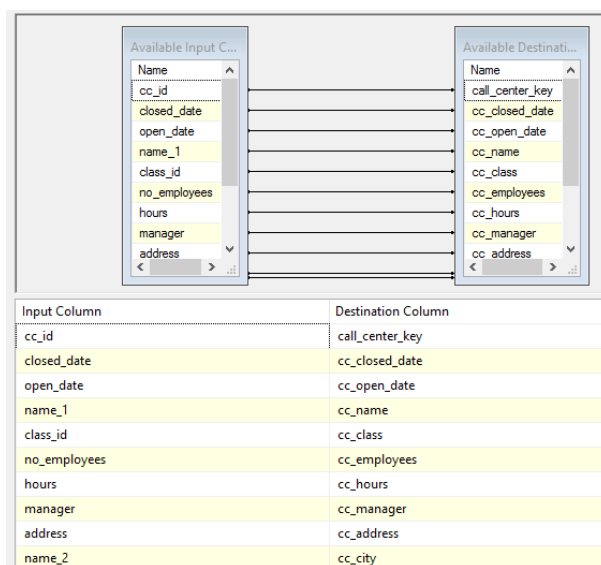
Primer *Select* SQL koda za ekstrakciju i transformaciju izvornih podataka koji se nalazi na *OLE DB Source* komponenti je prikazan na slici 4.

OLE DB connection manager:
WIN-7JHUH2K7P6S.vmart
Data access mode:
SQL command
SQL command text:
<pre>SELECT cc.cc_id, cc.closed_date, cc.open_date, cc.name as name_1, cc.class_id, cc.no_employees, cc.hours, cc.manager, cc.address, c.name as name_2, s.name as name_3, r.name FROM vmart.dbo.call_center cc INNER JOIN vmart.dbo.city c ON cc.city_id = c.city_id INNER JOIN vmart.dbo.state s ON c.state_id = s.state_id INNER JOIN vmart.dbo.region r ON s.region_id = r.region_id ORDER BY cc.cc_id</pre>

Slika 4. *OLE DB Source* editor

U okviru procesa konfiguracije se prvo bira odgovarajući izvor baze podataka, što je u ovom slučaju OLTP baza. Prikazani upit selektuje podatke iz tabela povezane sa *call_center* tabelom, a rezultujuće kolone su izabrane iz tabela koje učestvuju u *Join* procesu.

U dugoj komponenti, OLE DB Destination, prvo se vrši izbor destinacione baze tj. OLAP baze kao i izbor tabele u koju će se podaci koji su ekstarkovani i transformisani, učitati. Imajući u vidu da se nazivi kolona u upitu nad izvornom tabelom mogu razlikovati od naziva kolona u destinacionoj tabeli, moguće je tačno definisati koja se izvorna kolona uparuje sa odgovarajućom destinacionom kolonom. Na slici 5 je prikazan konfiguracioni prozor za mapiranje kolona u *public_call_center_dimension* tabeli.



Slika 5. Mapiranje kolona za tabelu *public_call_center_dimension*

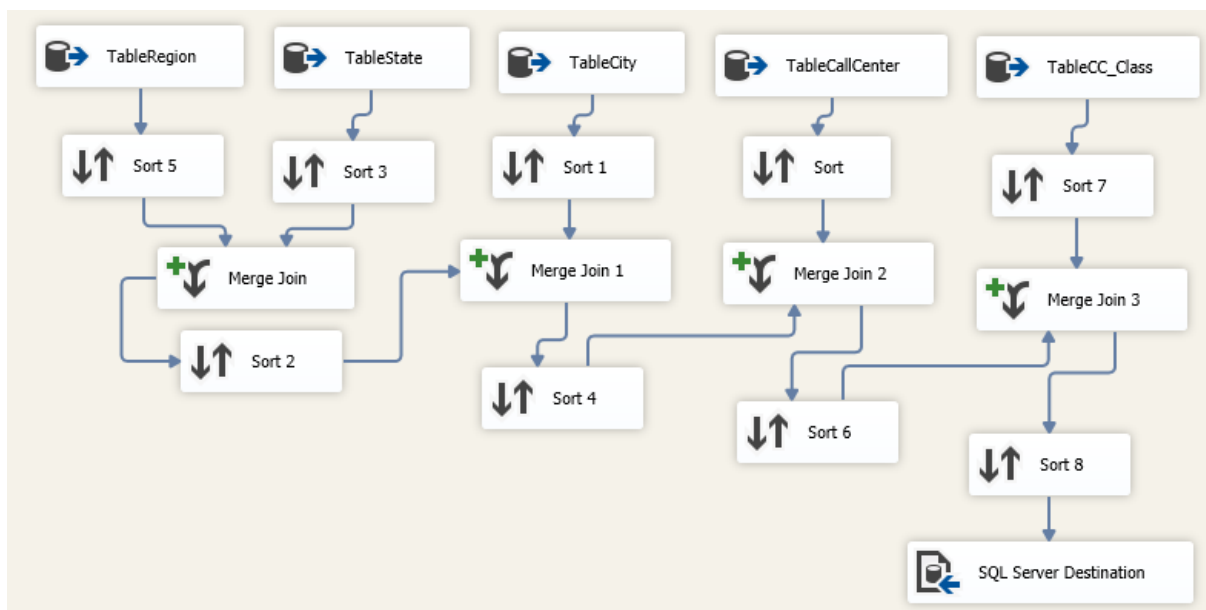
Za svaku destinacionu tabelu u pripremljenoj *Data Warehouse* bazi je dizajniran po jedan *job*.

5.1.2. ETL Rešenje 2 - korišćenje SSIS funkcionalnih komponenti

Za kreiranje ETL poslova se mogu koristiti ugrađene funkcionalnosti koje se nalaze u okviru kutije alata u SSIS menadžment konzoli. Jednostavnim postavljanjem i konfigurisanjem odgovarajućih dostupnih alata, bez potrebe poznavanja SQL programskog jezika se može dizajnirati svaki od potrebnih poslova. Kada poredimo ovaj pristup sa prethodno opisanim, možemo zaključiti da se ETL rešenje 1 izvršava na samoj bazi podataka dok se ETL rešenje 2 izvršava u okviru ETL aplikacije. Možemo takođe da uočimo da se u ovom rešenju OLE DB Source komponenta u stvari tabela a ne upit, kao što je slučaj u ETL rešenju 1.

Korišćenjem ETL aplikacije proces spajanja tabela se izvodi korišćenjem dve OLE DB izvorne komponente, koje u suštini ekstrahuju podatke iz dve tabele odvojeno radi sortiranja komponenti. Svaka izvorna tabela mora da bude sortirana po kolonama koje učestvuju u procesu spajanja da bi se rezultat sortirajućih komponenti spojio *Merge Join* komponentom. Kada postoji potreba da se izvrši nekoliko operacija spajanja tabela, potrebno je koristiti dve *Sort* i jednu *Merge Join* operaciju za svako spajanje. Za implementaciju ovog rešenja je korišćeno nekoliko ETL komponenti iz velikog broja dostupnih komponenti koje nudi SSIS.

Na slici 6 je prikazana šema komponenti ETL aplikacije koje su korišćene za ekstrakciju, transformaciju i učitavanje podataka u `public_call_center_dimension` tabelu.



Slika 6. Primer SSIS posla u Data Flow Editoru

Analizom prikazane šeme na slici 6 se može videti da je pet OLE DB Source komponenti korišćeno za pet izvornih tabela, kao i četiri *Merge Join* komponenti za izvršavanje *Inner-join* operacija. Da bi se podaci spojili *Merge Join* komponentom, neophodno je prethodno izvršiti sortiranje podataka izvornih tabela. Kako SSIS ne koristi indekse kolona, možemo očekivati slabije performanse izvođenja sortiranja u ovom slučaju.

Na slici 7 je predstavljen konfiguracioni prozor *Sort* komponente, na kojoj se mogu videti dostupne opcije, t.j. može se izabrati kolona, ili kolone po kojima će se izvršiti sortiranje. Potrebno je voditi računa pri selektovanju kolona u ovom prozoru, kako bi bile izabrane samo neophodne kolone, da ne bi došlo do učitavanja nepotrebnih podataka i time povećalo opterećenje i vreme za izvršenje procesa.

Available Input Columns		
☐	Name	Pass Through
☒	region_id	☐
☐	name	☒

Input Column	Output Alias
region_id	region_id

Slika 7. Sort Transformation Editor

Po podešavanju komponenti sortiranja, potrebno je u okviru *Merge Join* komponente izabrati i sve one kolone čiji će sadržaj biti uključen u izlazu ove komponente. Ovaj izbor treba napraviti u Merge Join Transformation Editor prozoru koji je prikazana na slici 8, izborom odgovarajućih polja sa leve strane izvornih tabela.

Join type: Inner join

Sort 5

☐	Name	Order	Join Key
☐	region_id	1	☒
☒	name	0	☐

Sort 3

☐	Name	Order	Join Key
☒	state_id	0	☐
☒	name	0	☐
☐	region_id	1	☒

Input	Input Column	Output Alias
Sort 5	name	name
Sort 3	name	name_1
Sort 3	state_id	state_id

Slika 8. Merge Join Transformation Editor

Poslednja komponenta prikazana u šemi ETL aplikacije je OLE DB Destination komponenta u okviru koje je potrebno izabrati destinacionu bazu i destinacionu tabelu. Ovo podešavanje je jednostavno izvesti koristeći funkcionalnost mapiranja kolona izvora i kolona u destinacionoj tabeli, slično kao što je predstavljeno u opisu ETL rešenja 1.

Kompletan posao (*control flow*) je prikazan na slici 9. Ikonice na slici su pojedinačne transformacije, a ikonica sa nazivom *call center dimension* na slici 9 predstavlja transformaciju prethodno prikazanu na slici 6.



Slika 9. Control Flow Editor za ceo ETL posao

5.1.3. ETL Rešenje 3 - korišćenje Pentaho ETL alata

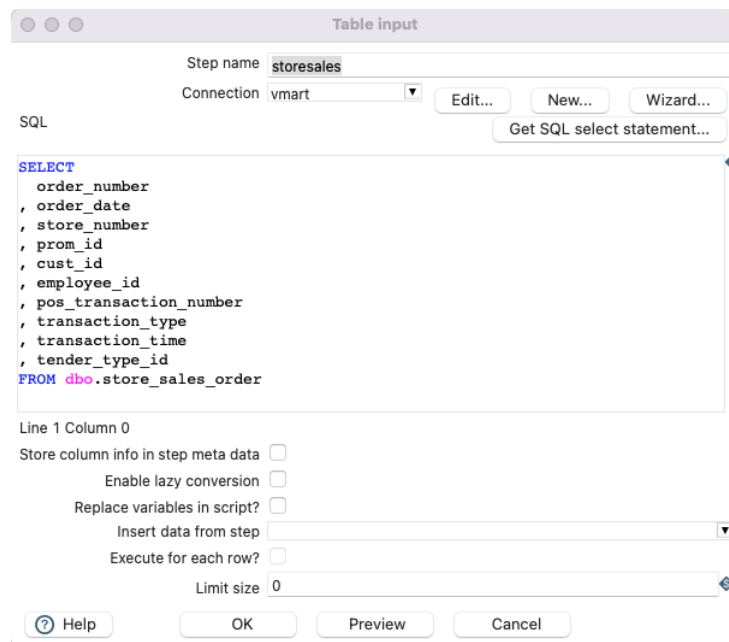
Pentaho Data Integration (PDI) Community Edition je besplatan ETL alat. Ovaj alat sadrži komponente za vizuelni dizajn procesnih tokova kao i konektore za povezivanje sa različitim izvorima podataka. U konkretnom slučaju, korišćen je PDI verzije 9.4 za čiji rad je preporučen Java SDK verzije 18. U cilju izvođenja eksperimenta, na zasebnoj virtualnoj mašini instaliran je operativni sistem Windows 10, Oracle Java SDK verzije 18 za Windows, PDI verzije 9.4 (<https://www.hitachivantara.com/en-us/products/pentaho-plus-platform/data-integration-analytics/pentaho-community-edition.html>), kao i Native MS SQL Server JDBC drajver (jtds-1.3.1.jar) koga je potrebno sačuvati u pod-direktorijumu lib ovog alata.

Pentaho Data Integration (PDI) nudi dve osnovne mogućnosti u svom okruženju:

1. **Job** – ovaj deo podrazumeva spajanje više transformacija kako bi one formirale jednu celinu i izvršile kompletan posao višestrukih transformacija i
2. **Transformation** – u njima se izvršavaju konkretne transformacije nad podacima (sortiranje, spajanje, dodavanje i uklanjanje kolona, punjenje destinacionih tabela, itd.).

Slično kao u rešenju 2, u rešenju 3 se koriste sledeće transformacije (koraci):

- **Table input** – ovde se bira konekcija ka bazi (u našem slučaju Vmart) i konkretna tabela iz koje se čitaju izvorni podaci. Na slici 10 je pokazan editor podešavanja ove transformacije.



Slika 10. Table input Editor

- **Sort rows** – ova transformacija služi da redove iz skupa podataka koji se nalaze na njenom ulazu poređa po rastućem / opadajućem redosledu jedne ili više kolona, pri čemu se u editoru transformacije može birati da li se sortiranje vrši vodeći računa o velikim / malim slovima. Na izlazu transformacije se nalazi isti broj kolona kao i na njenom ulazu, bez potrebe da se to decidivno navede. Editor transformacije je pokazan na slici 11, gde se vidi da podatke treba poređati u rastućem redosledu (Ascending = Y), ne vodeći računa o velikim i malim slovima (Case sensitive compare = N).

Step name:

Sort directory:

TMP-file prefix:

Sort size (rows in memory):

Free memory threshold (in %):

Compress TMP Files? ☒

Only pass unique rows? (verifies keys only) ☐

Fields :

#	Fieldname	Ascending	Case sensitive compare?	Sort based on current locale?	Collator Strength	Presorted?
1	order_number	Y	N	N	0	N

Slika 11. Sort rows Editor

- **Merge join** – ova transformacija omogućava povezivanje podataka iz dva izvora po uslovu koji se zadaje u editoru transformacije. Najčešći uslov je da su vrednosti u jednoj ili više kolona iz jednog izvora jednaki vrednostima iz jedne ili više kolona drugog izvora. Transformacija može sama da “otkrije” po kojim kolonama treba povezati podatke (opcija Get key fields), mada i korisnik može sam da definiše te kolone. Editor ove transformacije je pokazan na slici 12, gde se vidi da se na ulazu nalaze dva izvora: Sort rows i Sort rows 2 i da podatke iz ovih izvora treba spojiti po koloni order_number.

Step name:

First Step:

Second Step:

Join Type:

Keys for 1st step:

#	Key field
1	order_number

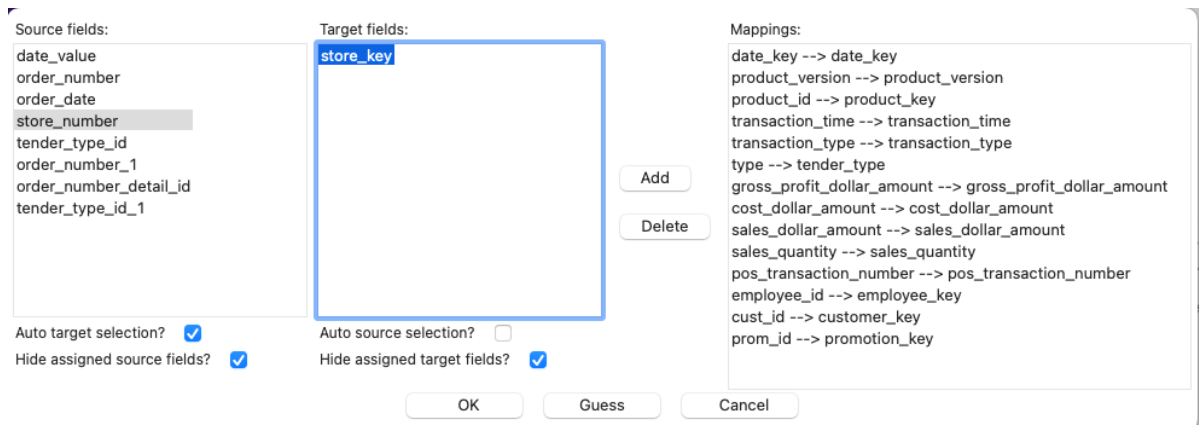
Keys for 2nd step:

#	Key field
1	order_number

Slika 12. Merge join Editor

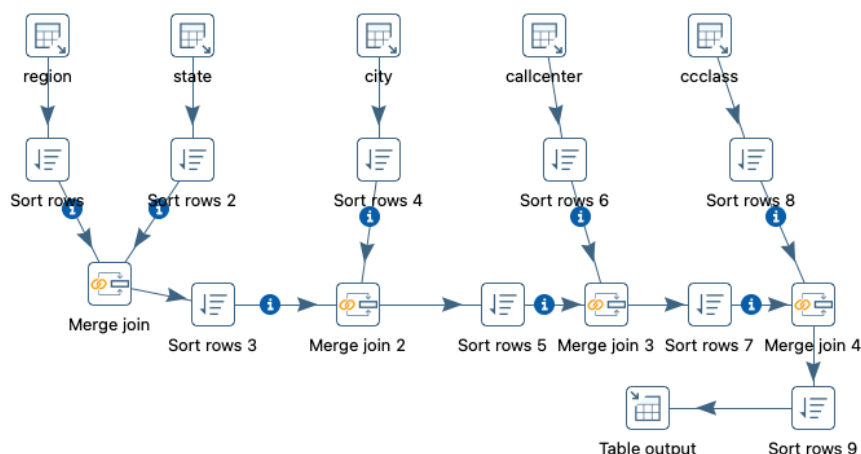
- **Table output** – ova transformacija omogućava učitavanje transformisanih podataka u destinacionu tabelu OLAP baze. Najčešće se prilikom *merge-join* transformacija u prethodnim koracima kreira veći broj kolona od potrebnih za destinacionu tabelu, tako da je u ovoj transformaciji potrebno mapirati koje kolone iz transformisanih podataka treba učitati u odgovarajuće kolone destinacione tabele. Na slici 13 je pokazan editor mapiranja podataka. Na slici se jasno vide kolone transformisanih

podataka u listi Source fields i kolone iz destinacione tabele u listi Target fields. Nakon izbora kolone iz jedne liste i kolone iz druge liste i klikom na dugme Add u listi Mappings se pojavljuje izabrano mapiranje.



Slika 13. Editor mapiranja kolona podataka nakon transformacija i destinacione tabele

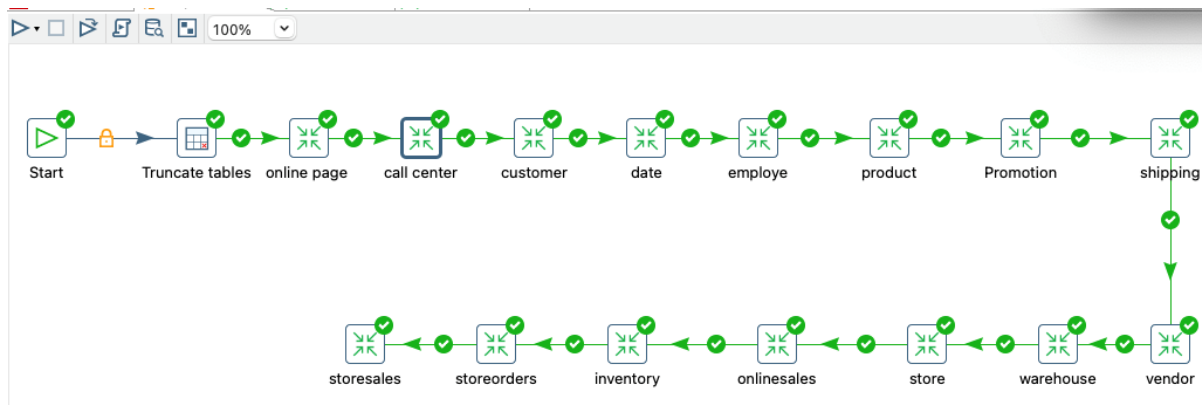
Kao i u rešenju 2, u ovom rešenju se transformacija podataka nekoliko tabela OLTP baze u jednu dimenzionu tabelu obavlja istim brojem koraka. Tako je na slici 14 prikazana transformacija ekvivalentna transformaciji u SSIS ETL alatu na slici 6.



Slika 14. Primer PDI transformacije 5 OLTP u jednu OLAP tabelu

Kako je cilj ETL posla da odgovarajući skup OLTP tabela transformiše u listu *dimension* i *fact* OLAP tabela, to je zadatak da se pojedinačne transformacije (kao što je ranije prikazano

pokazana na slici 9) poređaju i izvrše jedna za drugom. Najčešće se prvo obavljaju transformacije punjenja *Dimension* tabela, a zatim transformacije punjenja *fact* tabela. Kompletan posao (*job*) kreiran u Pentaho ETL alatu je pokazan na slici 15. Ikonica sa nazivom *call center* na slici 15 predstavlja transformaciju pokazanu na slici 14.



Slika 15. Izvršenje ETL posla transformacija OLTP tabela u listu Dimension i Fact OLAP tabela

5.2. Rezultati eksperimenata

Eksperimenti su koncipirani na ETL procesu koji se sastojao od nekoliko zadataka (tasks) i to:

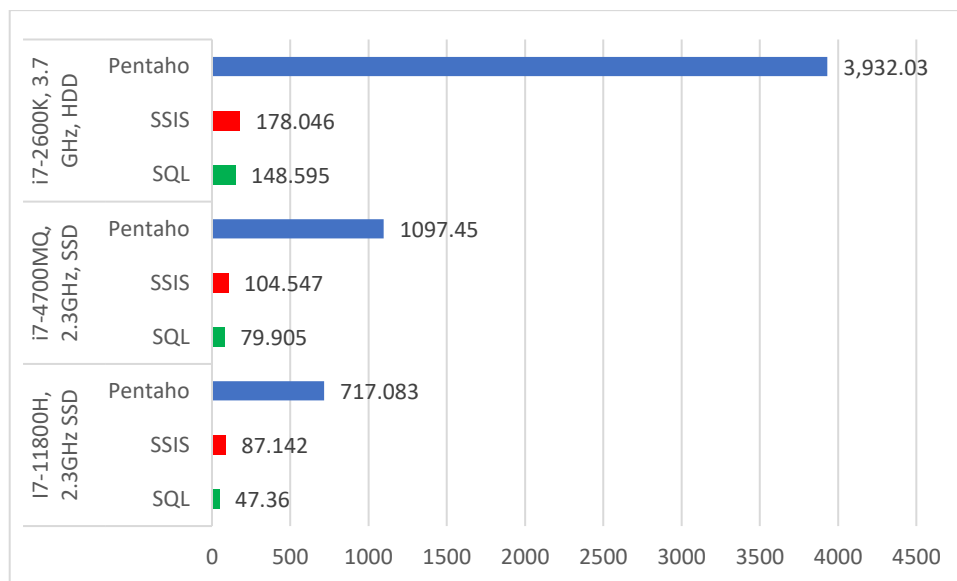
- Uklanjanje svih podataka iz tabela (truncating)
- Transformacija i učitavanje podataka u Dimenzione tabele (dva zadatka)
- Transformacija i učitavanje podataka u tabele činjenica (dva zadatka)

Posmatrano sa aspekta količine podataka u tabelama, devet od ukupno jedanaest dimenzionih tabela su sadržale manje od 2,000 zapisa (call_center_dimension, online_page_dimension, date_dimension, employee_dimension, promotion_dimension, shipping_dimension, vendor_dimension, warehouse_dimension, store_dimension) i ova grupa tabela je označena pod nazivom DimT1. Preostale dve dimenzione tabele su sadržale 50,000 i više zapisa (customer_dimension, product_dimension) i označene su pod nazivom DimT2. Tabele činjenica su podeljene u dve grupe, FactT1 u koju spadaju dve tabele (inventory_fact i store_orders_fact) sa po 300,000 zapisa i FactT2 u koju spadaju dve tabele (online_sales_fact i store_sales_fact) sa po 5,000,000 zapisa. U tabeli 4 su prikazana vremena izvršenja ETL poslova: Truncate, DimT1, DimT2, FactT1, FactT2 koristeći SQL kod, SSIS ETL i Pentaho ETL alate. Svaki posao je izvršen na tri različite hardverske platforme.

	i7-11800H, 2.3GHz SSD			i7-4700MQ, 2.3GHz, SSD			i7-2600K, 3.7 GHz, HDD		
	SQL	SSIS	Pentaho	SQL	SSIS	Pentaho	SQL	SSIS	Pentaho
truncate	0.079	0.047	0.198	0.109	0.094	0.547	0.156	0.156	3.653
DimT1	1.344	7.345	13.118	1.843	10.219	13.256	4.516	9.484	29.031
DimT2	0.89	3.609	8.347	1.344	5.156	10.587	2.907	8.39	25.819
FactT1	1.907	3.625	14.198	3.75	4.812	19.718	4.922	9.421	45.496
FactT2	43.14	72.516	681.222	72.859	84.266	1053.342	136.094	150.595	3,828.027
Total	47.36	87.142	717.083	79.905	104.547	1097.45	148.595	178.046	3,932.026

Tabela 4. Vreme izvršenja ETL poslova u sekundama

Na grafikonu 1 su prikazana ukupna vremena izvršenja svih ETL poslova za svaki od alata i za sve tri različite hardverske platforme.



Grafikon 4. ukupna vremena izvršenja svih ETL poslova za svaki od alata i za sve tri različite hardverske platforme u sekundama

Generalno posmatrano, kao što se vidi iz dobijenih rezultata eksperimenata, može se zaključiti da ukupno vreme izvršenja ETL poslova na datom primeru tabela nije preveliko, svega nekoliko minuta kod SQL koda i SSIS ETL alata, dok je kod Pentaho alata u rangi od 10 do 66 minuta. Takođe, rezultati pokazuju da je vreme izvršenja ETL poslova koristeći SQL kod i do 40% kraće u odnosu na ETL poslove koji se zasnivaju na SSIS aplikaciji sa adekvatnim sortiranjima i spajanjima podataka. Razlika u brzini izvršenja ETL procesa između SSIS sa jedne i Pentaho ETL alata sa druge je i preko 8 puta veća u korist SSIS. ETL poslovi su se najbrže izvršavali na računaru sa najboljim procesorom, kao što je i bilo očekivano.

5.3. Zaključak

Vreme potrebno za transformaciju i učitavanje podataka iz OLTP u OLAP tabele koje sadrže po nekoliko miliona redova je na nivou nekoliko minuta kod SQL koda i SSIS, dok je kod Pentaho ETL alata između 10 i 66 minuta. To ukazuje da se najbolji rezultati dobijaju kada se koristi ista platforma za bazu podataka, ETL alat i operativni sistem, zbog kompatibilnosti interfejsa. Moguće je da rezultati dobijeni navedenim eksperimentima mogu da se razlikuju u slučaju kada su platforme baza podataka i ETL rešenja postavljena na različitim računarima, zbog kašnjenja u mrežnom segmentu.

Imajući u vidu da je vreme potrebno za izvršavanje ETL poslova u slučajevima sa višemilionskim zapisima u tabelama prilično kratko, preporuka je da se ETL procesi pokreću češće radi izvršavanja poslovne analitike nad svežim podacima u OLAP bazi podataka.

Na izbor odgovarajućeg rešenja za poslovanje može uticati i cena samog rešenja. Microsoft SQL Server je podlozan licenciranju. SSIS komponenta je sastavni deo SQL Servera. U trenutku pisanja ove disertacije, na osnovu raspoloživih informacija na web sajtu proizvođača, Standard i Enterprise edicije SQL servera se licenciraju po *Per Core* i *Server + CAL (Client Access License)* modelima koristeći različite programe licenciranja. Cena licence SQL Server Enterprise edicije koja pokriva 2 procesorska jezgra kroz Volume Licensing kanal iznosi 15,123 USD. Sa druge strane Pentaho Data Integration (PDI) Community Edition je besplatan ETL alat.

U zavisnosti od raspoloživog budžeta i prihvatljivih performansi različitih ETL alata, poslovna preduzeća mogu doneti odluku o izboru odgovarajućeg alata. Mala i srednja preduzeća, ukoliko nemaju potrebu za transformacijom ogromne količine podataka, a ažuriranje OLAP baze se ne mora vršiti frekventno u toku dana, mogu izabrati besplatno ETL rešenje.

6. Performanse BI alata

Poslovna inteligencija (na engleskom *Business Intelligence* ili BI) predstavlja jedan od ključnih segmenata savremenog poslovnog okruženja. Ona predstavlja skup različitih metodologija i softverskih rešenja za analizu podataka. Kod izvođenja BI projekata je potrebno prvo izvršiti identifikaciju potrebnih istorijskih podataka. Oni mogu doći iz različitih izvora, kako iz internih transakcionih baza podataka preduzeća, tako i eksternih podataka koji mogu doći sa Internet mreže. Njihovo “čišćenje” kroz proces transformacije i učitavanje u OLAP bazu podataka se vrši uz pomoć ETL alata. OLAP baza podataka se naziva i *Data Warehouse* baza. Ona je optimizovana za brzo čitanje podataka i podaci su u njoj skladišteni po unapred dizajniranoj arhitekturi. BI alati se najčešće koriste od strane poslovnih analitičara i pružaju mogućnost različitih analiza nad podacima iz *Data Warehouse* baze podataka. Ovi alati su dizajnirani na način da ne zahtevaju tehnička znanja krajnjih korisnika. Krajnji korisnici mogu na jednostavan način da izaberu formate prikaza upita iz baze, upare podatke koji su međusobno zavisni, podese odgovarajuće filtere, izvrše različite tipove analiza i kreiraju odgovarajuće izveštaje. Unapred pripremljeni formati izveštaja potrebni za donošenje odluka mogu biti smešteni na interne portal lokacije u preduzeću. Na taj način im donosioci odluka mogu lako pristupiti i uz korišćenje dodatnih opcija u okviru samih pripremljenih izveštaja, izvršiti dodatna filtriranja i podešavanja radi uvida u informacije iz različitih perspektiva.

BI takođe uključuje procese poput analize poslovnih performansi, izveštavanja i upravljanja performansama. Kroz ove procese, preduzeća mogu identifikovati šablone, analizirati ključne pokazatelje performansi (KPI – *Key Performance Indicators*) i optimizovati operacije.

Prednosti korišćenja BI su višestruke. Poslovna preduzeća na ovaj način stiču sposobnost brze reakcije na promene u poslovnom okruženju, mogu proaktivno identifikovati nove prilike za rast i povećanje profita, kao i smanjiti troškove i unaprediti ukupnu efikasnost poslovanja. Zbog mogućnosti predviđanja budućih trendova na osnovu prediktivne analize istorijskih podataka, preduzeća mogu pre svoje konkurencije pokrenuti odgovarajuće aktivnosti radi sticanja komparativne prednosti na tržištu.

Kada se razmatra budućnost poslovne inteligencije, ona će nastaviti da evoluirati. Očekuje se da će tehnologije veštačke inteligencije (*Artificial Intelligence* - AI) i mašinskog učenja (*Machine Learning*) postati sastavni deo BI, pružajući još sofisticiranije analize i automatsko otkrivanje uzoraka. Takođe, mobilni pristup BI alatima se konstantno unapređuje, omogućavajući donosiocima odluka da pristupe ključnim informacijama u pokretu.

Može se zaključiti da BI nije samo alat, već je i ključni resurs za donošenje odluka na osnovu informacija izvedenih iz podataka u dinamičnom poslovnom okruženju. Njegova integracija u strategiju poslovanja donosi poboljšanje performansi i konkurentске prednosti na današnjem globalnom tržištu.

Cilj ove studije je poređenje performansi dva BI alata i to komercijalnog alata Microsoft PowerBI i *Open Source/besplatnog* alata Pentaho, prilikom izvođenja istih tipova upita nad unapred pripremljenom *Data Warehouse* bazom VmartD, postavljenom na SQL Serveru 2017. Imajući u vidu da brzina izvršavanja BI upita zavisi od performansi računara na kojima se oni izvršavaju, eksperimenti su vršeni na nekoliko računara različitih hardverskih karakteristika.

Eksperimenti su izvedeni na Microsoft SQL Server SUPB sa Microsoft Imagine licencom.

6.1. Okruženje za izvođenje eksperimenata

Data Warehouse baza podataka VmartD koja je korišćena u predstavljenoj ETL studiji u poglavlju 5 ovog rada, je korišćena i za potrebe eksperimenata poređenja performansi BI alata.

VmartD baza je postavljena na VMware virtualnoj mašini sa instaliranim Microsoft Windows 2016 Server Standard operativnim sistemom u okviru Microsoft SQL Servera 2017. Pri instalaciji okruženja je korišćena licenca za obrazovne ustanove. Virtuelnoj mašini je dodeljeno 6 GB RAM i 120GB prostora na disku.

Posebne virtualne mašine sa odgovarajućim BI alatima su pripremljene za potrebe eksperimenata pokretanja upita ka VmartD bazi podataka i to:

- Virtualna mašina sa instaliranim Windows 10 Pro operativnim sistemom i Microsoft PowerBI desktop aplikacijom, kojoj je dodeljeno 4 GB RAMa i 60 GB prostora na disku.
- Virtualna mašina sa instaliranim Windows 10 Pro operativnim sistemom i besplatnim Pentaho Business Analytics Community Edition alatom, kojoj je dodeljeno 4 GB RAM-a i 60 GB prostora na disku.

Obe virtualne mašine sa BI alatima su povezane sa VmartD bazom podataka koristeći VMware virtualne mrežne interfejsa sa NAT podešavanjem.

Dizajnirana su 4 različita upita ka VmartD bazi za izvođenje eksperimenata i implementirani su koristeći odgovarajuće tabelarne i grafičke interfejsa u okviru oba BI alata. Struktura upita je predstavljena u daljem tekstu, koristeći SQL jezik.

Upit 1

Postavljeni cilj Upita 1 (Q1) je testiranje performansi upita i prikaza rezultata upita u jednoj tabeli činjenica *online_sales_fact* od 5 miliona redova, spojenoj *inner-join* vezom sa jednom kolonom u dimenzionoj tabeli *public_date_dimension* koja se sastoji od 1,826 redova. Polje vrednosti profita USD valute je korišćeno za sumiranje ukupne vrednosti ostvarenog profita, grupisane po koloni godišnjih kvartala iz navedene dimenzione tabele i rastućim redosledom vremenske dimenzije. Definisani upit treba da prikaže rezultate ostvarenog profita onlajn prodaje po godišnjim kvartalima. Format prikaza je tabelarni u oba BI alata.

```

SELECT sum(o.gross_profit_dollar_amount) AS sum_profit,
d.calendar_year_quarter
FROM online_sales_fact o
INNER JOIN public_date_dimension d ON
    o.sale_date_key = d.date_key
GROUP BY d.calendar_year_quarter
ORDER BY d.calendar_year_quarter

```

Upit 2

Upit 2 (Q2) je sličan prethodnom upitu i dizajniran je za mali skup rezultatnih podataka radi prikazivanja u vizualnom formatu stubova. Upit se vrši nad tabelom činjenica *online_sales_fact* od 5 miliona redova, koja je spojena *inner-join* vezom sa kolonom *date_key* u dimenzionoj tabeli *public_date_dimension*. Rezultat upita treba da prikaže ukupan profit svih prodajnih transakcija grupisan po godinama i rastućim redosledom vremenske dimenzije.

```

SELECT sum(o.gross_profit_dollar_amount) AS sum_profit, d.calendar_year
FROM online_sales_fact o
INNER JOIN public_date_dimension d ON
    o.sale_date_key = d.date_key
GROUP BY d.calendar_year
ORDER BY d.calendar_year

```

Upit 3

Ovaj upit je dizajniran sa ciljem vraćanja većeg seta rezultatnih podataka, koji se kasnije u samom BI alatu mogu dodatno filtrirati.

Upit se vrši nad tabelom činjenica *online_sales_fact* od 5 miliona redova koja je spojena *inner-join* vezama sa dve dimenzione tabele, *store_dimension* i *public_date_dimension* preko kolona *store_key* i *date_key*, respektivno. Cilj upita je da prikaže ukupnu vrednost ostvarenog profita, grupisanog po nazivu prodavnica, gradova i regiona u kojima se prodavnice nalaze kao i godišnjim kvartalima. Postavljeni redosled je rastući, po kolonama naziva prodavnica i godišnjim kvartalima. Format prikaza rezultata je tabelarni.

```

SELECT sum(ssf.gross_profit_dollar_amount) AS sum_profit, s.store_name,
s.Store_City, s.store_region, d.calendar_year_quarter
FROM store_sales_fact ssf
INNER JOIN store_dimension s ON
    ssf.store_key = s.store_key
INNER JOIN public_date_dimension d ON
    ssf.date_key = d.date_key
GROUP BY s.store_name, s.Store_City, s.store_region,
d.calendar_year_quarter
ORDER BY s.store_name, d.calendar_year_quarter;

```

Upit 4

Upit 4, slično kao i prethodni, ima za cilj vraćanje većeg seta rezultatnih podataka.

Izvršenje upita je nad tabelom činjenica *store_orders_fact* koja sadrži 300,000 zapisa transakcija porudžbina u radnjama i koja je spojena *inner-join* vezama sa dve dimenzione tabele, *public_employee_dimension* i *public_vendor_dimension* preko kolna *employee_key* i *vendor_key*, respektivno. Cilj upita je da prikaže ukupnu vrednost ostvarenih porudžbina, grupisanu po titulama zaposlenih koji su ostvarili prodaju i proizvođačima poručene robe.. Redosled prikaza rezultata je opadajući po ukupnoj vrednosti porudžbina i rastući po titulama zaposlenih i proizvođačima poručene robe. Rezultati upita će biti prikazani u tabelarnom formatu u oba BI alata.

```

SELECT sum(sof.total_order_cost) as Total_Order_Value, e.job_title,
v.vendor_name
FROM store_orders_fact sof
INNER JOIN public_employee_dimension e ON
    sof.employee_key = e.employee_key
INNER JOIN public_vendor_dimension v ON
    sof.vendor_key = v.vendor_key
GROUP BY e.job_title, v.vendor_name
ORDER BY Total_Order_Value DESC, e.job_title, vendor_name;

```

Za svaki upit će biti izvršena po tri merenja u svakom BI alatu i prosečna vrednost rezultata će biti prikazana kao rezultatna.

Radi uspostavljanja konekcije između obe virtualne mašine konfigurisan je Windows firewall na SQL server virtualnoj mašini da propusti svu komunikaciju ka i od virtualnih mašina sa BI alatima.

Za potrebe izvršenja eksperimenata izabrane su tri hardverske platforme na kojima su postavljene virtualne mašine:

- Računar sa I7-11800H, 2.3GHz procesorom i SSD diskom,
- Računar sa i7-9750H, 2.6GHz procesorom i SSD diskom,
- iMac računar (21.5-inch, 2017) sa i5-7360U 2.3GHz procesorom.

6.1.1. BI rešenje 1 – Power BI Desktop

Power BI Desktop predstavlja Microsoft BI alat koji omogućava analizu i vizualizaciju podataka dobijenih iz različitih izvora. On se može besplatno preuzeti i instalirati na računaru pod Windows operativnim sistemom. Power BI Desktop omogućava povezivanje na više različitih izvora podataka koji se mogu kombinovati, odnosno modelovati u odgovarajući data model. Takav data model se može koristiti za kreiranje različitih vizualizacija podataka ili kolekcija vizualizacija koje se mogu podeliti sa drugim korisnicima u organizaciji. Najčešće se izveštaji kreiraju od strane poslovnih analitičara, a potom se dele sa drugim korisnicima koristeći Power BI servis. Power BI servis podleže pravilima licenciranja. Za potrebe vršenja eksperimenata u ovoj studiji je korišćena samo besplatna Power BI desktop aplikacija.

U Power BI Desktop konzoli su kreirana sva četiri prethodno definisana upita.

Upit 1 je kreiran na sledeći način. Izborom *Get Data* opcije je izabran SQL Server kao izvor podataka, upisana je IP adresa servera i izabran je *Direct Query* režim povezivanja. U novootvorenom prozoru Navigator su izabrane *online_sales_fact* i *public_date_dimension* tabele VmartD baze podataka, kao što je prikazano na slici 16.

Navigator

Display Options ▾

- ▶ TestD
- ▶ vmart
- ▲ vmartd [16]
 - ☐ online_page_dimension
 - ☒ online_sales_fact
 - ☐ public_call_center_dimension
 - ☐ public_customer_dimension
 - ☒ public_date_dimension
 - ☐ public_employee_dimension
 - ☐ public_inventory_fact
 - ☐ public_online_page_dimension
 - ☐ public_product_dimension
 - ☐ public_promotion_dimension
 - ☐ public_shipping_dimension
 - ☐ public_vendor_dimension
 - ☐ public_warehouse_dimension
 - ☐ store_dimension
 - ☐ store_orders_fact
 - ☐ store_sales_fact
- ▶ vmartd1

public_date_dimension

date_key	date	full_date_description	day_of_week	day
1	1/1/2003 12:00:00 AM	January 1, 2003	Wednesday	
2	1/2/2003 12:00:00 AM	January 2, 2003	Thursday	
3	1/3/2003 12:00:00 AM	January 3, 2003	Friday	
4	1/4/2003 12:00:00 AM	January 4, 2003	Saturday	
5	1/5/2003 12:00:00 AM	January 5, 2003	Sunday	
6	1/6/2003 12:00:00 AM	January 6, 2003	Monday	
7	1/7/2003 12:00:00 AM	January 7, 2003	Tuesday	
8	1/8/2003 12:00:00 AM	January 8, 2003	Wednesday	
9	1/9/2003 12:00:00 AM	January 9, 2003	Thursday	
10	1/10/2003 12:00:00 AM	January 10, 2003	Friday	
11	1/11/2003 12:00:00 AM	January 11, 2003	Saturday	
12	1/12/2003 12:00:00 AM	January 12, 2003	Sunday	
13	1/13/2003 12:00:00 AM	January 13, 2003	Monday	
14	1/14/2003 12:00:00 AM	January 14, 2003	Tuesday	
15	1/15/2003 12:00:00 AM	January 15, 2003	Wednesday	
16	1/16/2003 12:00:00 AM	January 16, 2003	Thursday	
17	1/17/2003 12:00:00 AM	January 17, 2003	Friday	
18	1/18/2003 12:00:00 AM	January 18, 2003	Saturday	
19	1/19/2003 12:00:00 AM	January 19, 2003	Sunday	
20	1/20/2003 12:00:00 AM	January 20, 2003	Monday	
21	1/21/2003 12:00:00 AM	January 21, 2003	Tuesday	
22	1/22/2003 12:00:00 AM	January 22, 2003	Wednesday	
23	1/23/2003 12:00:00 AM	January 23, 2003	Thursday	

Select Related Tables

Load

Transform Data

Cancel

Slika 16. Navigator prozor za izbor tabela iz baze podataka

Radi kreiranja relacija između navedenih tabela korišćena je opcija *Manage relationships* sa *Modeling* ribona. Tabele su povezane relacijom *Many to one* koristeći *Sale_Date_Key* i *Date_Key* kolone, kao što je prikazano na slici 17.

Create relationship

Select tables and columns that are related.

online_sales_fact

sale_date_key	ship_date_key	product_key	product_version	customer_key	call_center_key	online_pa
1015	1018	2715	1	41771	162	
1775	1779	12107	1	6240	184	
677	682	3247	1	10010	154	

public_date_dimension

date_key	date	full_date_description	day_of_week	day_number_in_calendar_month	day_r
121	5/1/2003 12:00:00 AM	May 1, 2003	Thursday		1
122	5/2/2003 12:00:00 AM	May 2, 2003	Friday		2
125	5/5/2003 12:00:00 AM	May 5, 2003	Monday		5

Cardinality

Many to one (*:1)

Cross filter direction

Single

☒ Make this relationship active

☐ Apply security filter in both directions

☐ Assume referential integrity [Learn more](#)

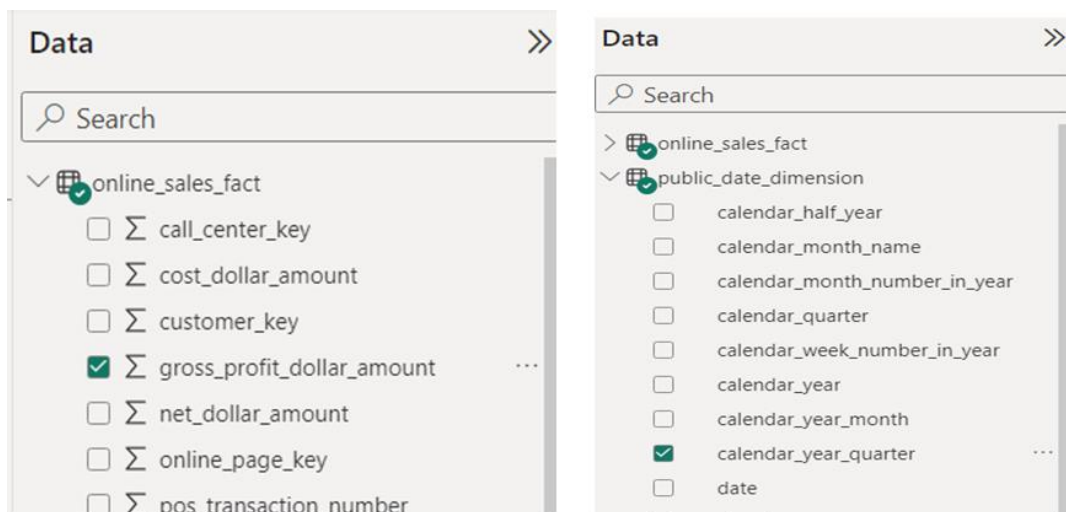
OK

Cancel

Slika 17. Kreiranje relacija između tabela u prozoru Create Relationship

Iz sekcije sa vizualizacijama je izabran tabelaran prikaz izveštaja.

U sekciji Data su izabrana polja/kolone izveštaja, $\Sigma gross_profit_dollar_amount$ iz tabele *online_sales_fact* i *calendar_year_quarter* iz tabele *public_date_dimension*, kao što je prikazano na slici 18.



Slika 18. Izbor kolona izveštaja

Upit 2 je definisan na sličan način, izborom istih tabela kao u Upitu 1, pri čemu je u sekciji sa vizualizacijama izabran *Clustered Column Chart* prikaz izveštaja. U sekciji Data su izabrana polja/kolone izveštaja, $\Sigma gross_profit_dollar_amount$ iz tabele *online_sales_fact* i *calendar_year* iz tabele *public_date_dimension*.

Za konfiguraciju upita 3 su izabrane tabele *store_sales_fact*, *store_dimension* i *public_date_dimension*. Kreirane su dve relacije između tabela. Prva relacija, *Many to one* je definisana između tabela *store_sales_fact* i *store_dimension* koristeći kolone *store_key* u obe tabele. Druga relacija, *Many to one* je definisana između tabela *store_sales_fact* i *public_date_dimension* koristeći kolone *date_key* u obe tabele. Iz sekcije sa vizualizacijama je izabran tabelaran prikaz izveštaja. Selektovane su sledeće kolone izveštaja: $\Sigma gross_profit_dollar_amount$ iz tabele *store_sales_fact*; *store_name*, *store_city* i *store_region* iz tabele *store_dimension*; *calendar_year_quarter* iz tabele *public_date_dimension*.

Prilikom konfiguracije upita 4 su izabrane tabele *store_orders_fact*, *public_employee_dimension* i *public_vendor_dimension*.

Kreirane su dve relacije između navedenih tabela. Prva relacija, *Many to one* je definisana između tabela *store_orders_fact* i *public_employee_dimension* koristeći kolone *employee_key* u obe tabele. Druga relacija, *Many to one* je definisana između tabela *store_orders_fact* i *public_vendor_dimension* koristeći kolone *vendor_key* u obe tabele. Iz sekcije sa vizualizacijama je izabran tabelaran prikaz izveštaja. Za kolone izveštaja su izabrane $\Sigma total_order_cost$ iz tabele *store_sales_fact*, *job_title* iz tabele *public_employee_dimension* i *vendor_name* iz tabele *public_vendor_dimension*.

Svaki od navedenih upita je sačuvan u okviru zasebnog fajla na hard disku.

Power BI Desktop poseduje ugrađenu komponentu Performance Analyzer, koji se koristiti za merenje vremena izvršenja i prikaza upita. Izborom opcije Start Recording i potom Refresh se može videti vreme izvršenja upita u ms. Da bi se izbeglo korišćenje keširanih podataka iz ranijih upita, pre svakog merenja iz menija File – Options and Settings – Options, u sekciji Data Load, će se isprazniti sačuvani keš podaci, kao što je prikazano na slici 19.

Data Cache Management Options ⓘ

Currently used: 585 KB

Clear Cache

Maximum allowed (MB):

4096 ⓘ

Restore Defaults

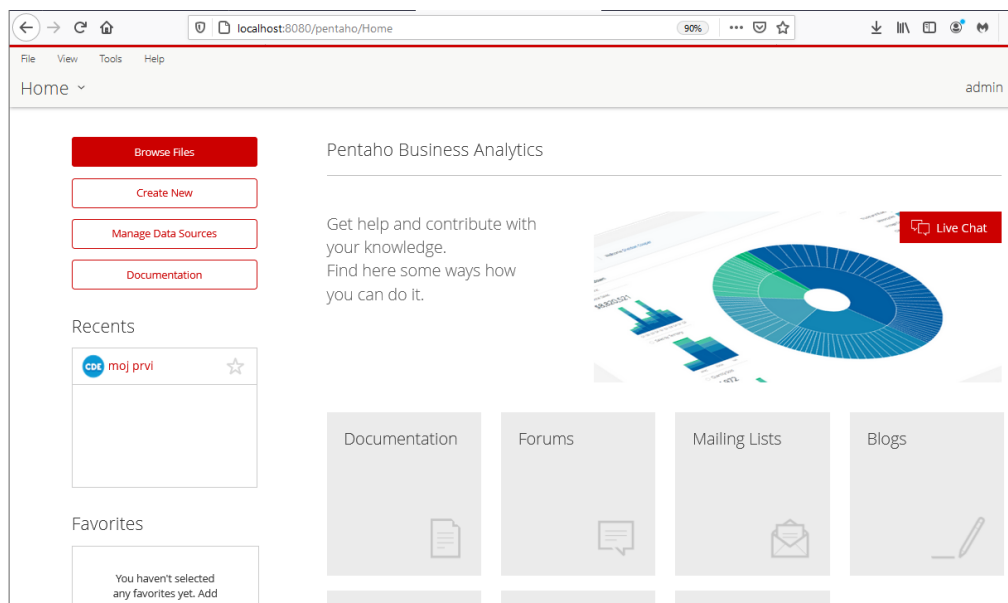
Slika 19. Pražnjenje keša iz Data Cache management opcija

6.1.2. BI Rešenje 2 – Pentaho Analytics

Pentaho predstavlja softversku platformu koja se koristi za rad sa podacima, počevši od pristupa, integracije, manipulacije do vizualizacije i analize podataka. Podaci mogu biti smešteni na različitim izvorima uključujući pored tradicionalnih baza podataka i izvore kao što su Hadoop klasteri, NoSQL baze i skladišta u *cloud*-u. Pentaho Analytics je komponenta Pentaho platforme koja se koristi za analizu i vizualizaciju podataka. Za pripremu jednostavnijih izveštaja nisu potrebne programerske veštine. Ukoliko je potrebno dizajnirati kompleksnije izveštaje, Pentaho nudi mogućnost korišćenja aplikativnih programskih interfejsa, mada su u tom slučaju potrebne izvesne programerske veštine.

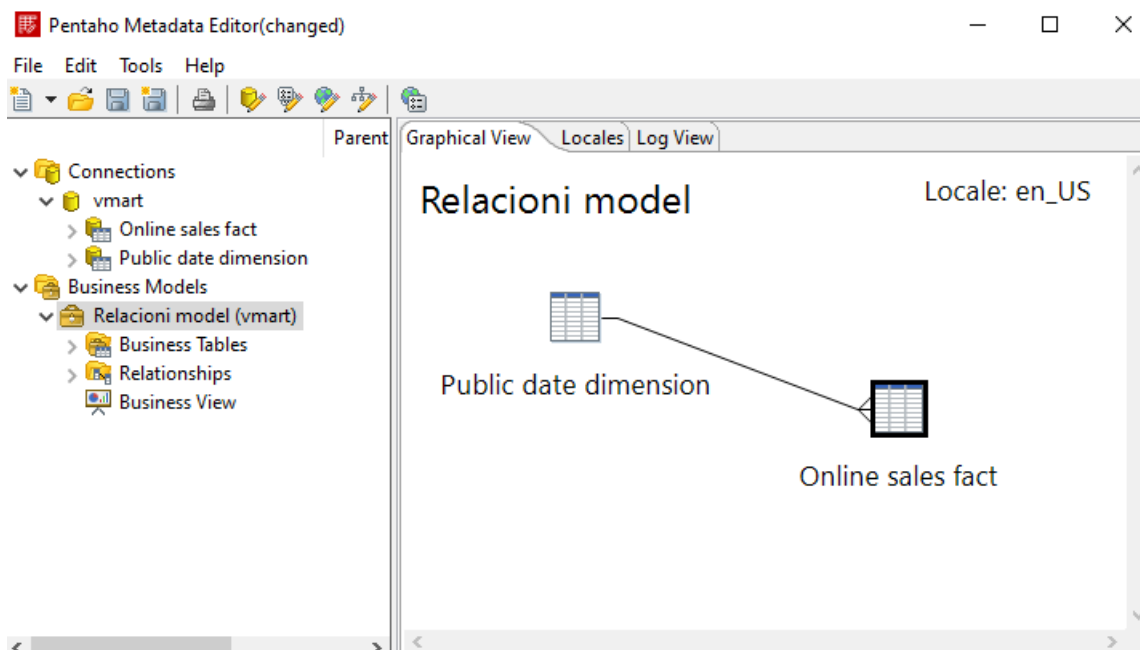
Pentaho softverska platforma dolazi u dve edicije, besplatna Pentaho Community edicija i Pentaho Enterprise edicija bazirana na pretplati. Za potrebe izvršenja eksperimenata je korišćena Pentaho Community edicija instalirana na Windows 10 Pro operativnom sistemu.

Na virtualnoj mašini sa Windows 10 operativnim sistemom je izvršena prethodna priprema koja uključuje pokretanje Web aplikacionog servera Tomcat. Početna web strana Pentaho Business Analytics je prikazana na slici 20.



Slika 20. Početna web strana Pentaho Business Analytics

Alat Pentaho Metadata editor je korišćen za kreiranje konekcije ka VmartD bazi podataka kao i definisanje relacionih modela za svaki od 4 predmetna izveštaja. Pogled na Pentaho Metadata editor sa kreiranim relacionim modelom za Upit 1 je prikazan na slici 21, dok je kreiranje relacija između tabela koristeći Pentaho Metadata Editor prikazano na slici 22.



Slika 21. Pentaho Metadata Editor za definisanje Metadata Source u Pentaho BI

Relationship Properties

From Table / Field: BT ONLINE SALES FACT ONLINE SALES FACT

To Table / Field: BC ONLINE SALES FACT SALE DATE KEY

Relationship: N:1

Join type: Inner

Complex Join?: ☐

Complex Join Metadata Formula:

Description:

OK Cancel

Slika 22. Kreiranje relacija između tabela u prozoru Create Relationship

Upiti su na Pentaho BI serveru kreirani direktno, upisom SQL naredbi nad već snimljenom konekcijom (*datasource* konekcija kreirana u metadata editoru) u polju *Query* u *Datasource* Panelu BI servera. Primer upisa upita 1 je prikazan na slici 23.

Sql Editor

Functions +

Parameters +

1 SELECT sum(o.gross_profit_dollar_amount) AS sum_profit,
2 d.calendar_year_quarter
3 FROM online_sales_fact o
4 INNER JOIN public_date_dimension d ON
5 o.sale_date_key = d.date_key
6 GROUP BY d.calendar_year_quarter
7 ORDER BY d.calendar_year_quarter
8

Ok Cancel

Slika 23. Upis upita u polju Query u Datasource Panelu

Pentaho Business Analytics server ne nudi ugrađenu mogućnost prikazivanje vremena izvršenja procesa, od trenutka pokretanja upita ka bazi podataka, do prikaza izveštaja na web stranici. Da bi se izmerilo vreme trajanja ovog procesa korišćen je sledeći pristup: potrebno vreme da Pentaho Business Analytics prikaže web stranicu sa agregiranim podacima na osnovu zadatog upita se dobija oduzimanjem sistemskog vremena u trenutku u kada je izveštaj prikazan i sistemskog vremena u trenutku kada je pokrenut upit ka serveru baza podataka. Radi merenja vremena izvršenja kompletnog procesa kreirana je Java skripta. JavaScript koji pokazuje sistemsko vreme početka procesa u pasusu pod nazivom “datetime1” je prikazan na slici 24.

```

1 <h2>upit 1</h2>
2 <p id="datetime1"></p>
3 <p id="datetime2"></p>
4
5 <script>
6     var now1=new Date();
7     var dt1=now1.toISOString();
8     document.getElementById("datetime1").innerHTML="vreme pocetka: "+dt1;
9 </script>

```

Slika 24. JavaScript koji prikazuje sistemsko vreme započetog procesa

JavaScript koji prikazuje sistemsko vreme završetka kreiranja izveštaja u pasusu pod nazivom “datetime2” je prikazan na slici 25.

```

1 function F2()
2 {
3     var now2=new Date();
4     var dt2=now2.toISOString();
5     document.getElementById("datetime2").innerHTML="vreme zavrsetka: "+dt2;
6 }
7

```

Slika 25. JavaScript koji prikazuje sistemsko vreme završetka izveštaja

6.2. Rezultati eksperimenata

Imajući u vidu da Power BI Desktop predstavlja desktop aplikaciju koja se koristi za prikaz izveštaja, a Pentaho Business Analytics koristi web serversku komponentu baziranu na Javi i instaliranu na nivou operativnog sistema za generisanje izveštaja u okviru web stranice, inicijalna pretpostavka je da će Pentahu biti potrebno više vremena za izvođenje operacija upita i prikaza izveštaja u odnosu na Power BI desktop. Nepoznanica pre izvođenja eksperimenata je da li operativni sistemi koji se koriste na Host računarima, a koji su bazirani na Unixu, kao što je MacOS mogu da utiču na brže izvršenje procesa na virtuelnoj mašini sa Pentaho rešenjem u odnosu na Host računare sa Windows operativnim sistemom i virtualnom mašinom sa PowerBI desktop rešenjem.

Pored primarnog cilja istraživanja koji se ogleda u merenju i poređenju performansi oba alata za koji se pretpostavlja da će Power BI desktop biti brži, sekundarni cilj je utvrđivanje nivoa eventualnog vremenskog kašnjenja zbog korišćenja web servisa na Pentaho Virtualnoj mašini.

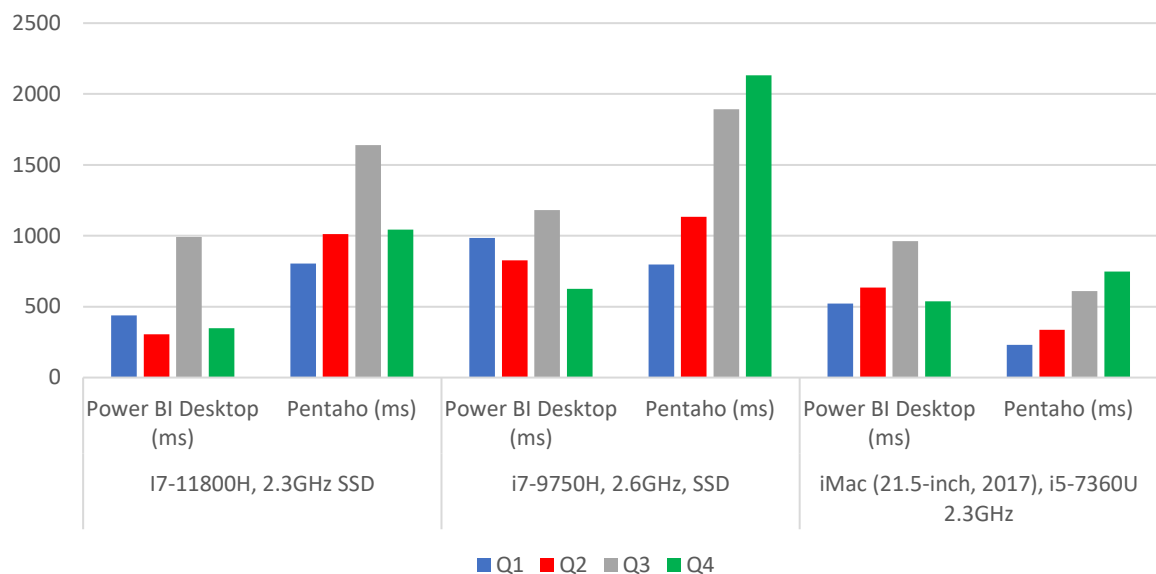
U tabeli 5 su prikazani rezultati merenja brzine upita. Tabela je formatirana za prikazivanje više dimenzija i to:

- Arhitektura računara,
- Korišćeni BI alat i
- Broj/oznaka upita (Q1-Q4)

	i7-11800H, 2.3GHz SSD		i7-9750H, 2.6GHz, SSD		iMac (21.5-inch, 2017), i5-7360U 2.3GHz	
	Power BI Desktop (ms)	Pentaho (ms)	Power BI Desktop (ms)	Pentaho (ms)	Power BI Desktop (ms)	Pentaho (ms)
Q1	438	804.33	984	798.33	521	231.33
Q2	304.67	1012.67	825.67	1134.67	634	335.67
Q3	991.33	1639	1181.33	1892.67	963	610.67
Q4	348.67	1044	626.33	2131.33	537.33	747.33

Tabela 5. Rezultati eksperimenata BI alata prikazani u milisekundama

Na grafikonu 5 su predstavljeni rezultati eksperimenata BI alata.



Grafikon 5. Rezultati eksperimenata BI alata prikazani u milisekundama

Na osnovu dobijenih rezultata se može videti da su na I7-11800H hardverskoj konfiguraciji sva četiri upita izvršena na PowerBI desktopu bila brža u odnosu na Pentaho za 1.65 puta (kod upita Q3) do 3.3 puta (kod upita Q2). Prednost u brzini izvršenja upita koristeći PowerBI desktop na hardverskoj konfiguraciji i7-9750H je bila u sličnom rangui u odnosu na Pentaho i kretala se u odnosu 1.37 (kod upita Q2) do 3.40 (kod upita Q4), sa izuzetkom brzine izvršenja upita Q1 gde je Pentaho bio brži u odnosu na PowerBI desktop za 1.23 puta. Obe prethodno navedene hardverske platforme su koristile Windows 10 operativni sistem na host računaru.

Rezultati na MacOS hardverskoj platformi su pokazali brže izvršenje upita Pentaho rešenja u odnosu na PowerBI Desktop za čak 2.5 puta kod upita Q1, 1.88 puta kod upita Q2 i 1.58 puta kod upita Q3, dok je upit Q4 bio brži kod PowerBi desktop rešenja za 1.39 puta.

6.3. Zaključak

Posebno treba imati na umu da je baza podataka postavljena na Microsoft platformi operativnog sistema (Windows Server 2016) i Microsoft platformi baze podataka (SQL Server 2017), kao i da su sve virtualne mašine sa BI alatima instalirane na Microsoft operativnom sistemu, Windows 10. Na osnovu rezultata eksperimenata možemo izvesti nekoliko zaključaka:

1. Na najsnažnijoj hardverskoj konfiguraciji na kojoj je host računar Windows 10, PowerBI Desktop je pokazao značajno bolje rezultate od Pentaho alata.
2. Na slabijoj hardverskoj konfiguraciji na kojoj je host računar Windows 10, PowerBI Desktop je pokazao bolje rezultate kod kompleksnijih upita dok je kod najmanje kompleksnog upita Pentaho imao bolji rezultat.
3. Na MacOS hardverskoj platformi je Pentaho generalno pokazao bolji rezultat u odnosu na PowerBI Desktop. Ovakav rezultat se može objasniti boljom kompatibilnosti MacOS-a i Pentaho Web servera, koji u pozadini koriste Unix sistem. Pentaho koristi web server zasnovan na Javi, koja se brže izvršava na Unix-baziranom sistemu.

Posmatrajući sve rezultate koristeći oba BI alata, najveće odstupanje rezultata je manje od 1 sekunde. Samim tim se može zaključiti da ta razlika u brzini, čak i kod najkompleksnijih upita ne predstavlja presudni uticaj na brzinu donošenja poslovnih odluka. Veći uticaj na izbor odgovarajućeg BI rešenja može biti cena svakog od rešenja.

Microsoft Power BI Desktop se može besplatno preuzeti i koristiti na računaru pod Windows operativnim sistemom. Ukoliko je potrebno da se izveštaji kreirani od strane poslovnih analitičara dele sa drugim korisnicima, potrebno je kupiti pretplatničku licencu za Power BI servis. U zavisnosti od potrebnih funkcionalnosti, u ponudi su Power BI Pro pretplata koja iznosi 10 USD po korisniku mesečno i Power BI Premium koja iznosi 20 USD po korisniku mesečno.

Pentaho platforma dolazi u dve edicije, besplatna Pentaho Community edicija i Pentaho Enterprise edicija bazirana na pretplati.

Imajući u vidu visok nivo integracije testiranih alata sa brojnim SUBP i da je razlika u performansama testiranih alata na nivou manjem od 1 sekunde, preporuka za poslovna preduzeća je da koriste BI rešenja koja omogućuju distribuciju izveštaja za poslovne korisnike i koja su besplatna.

7. Zaključak

U cilju analize performansi različitih sistema za upravljanje bazama podataka, različitih alata za izvođenje ETL procesa i različitih BI alata uloženo je mnogo truda i rada u instaliranje i podešavanje različitih kombinacija operativnih sistema, sistema za upravljanje bazama podataka, ETL i BI alata, u učitavanje podataka u instalirane baze koristeći različite SQL dijalekte i alate za učitavanje podataka, kao i izvođenje samih eksperimenata.

Instaliranje jedne kombinacije operativnog sistema i odgovarajućih SUBP, ETL i BI alata na nekoliko računara radi izvođenja eksperimenata, brisanje postojećih i instaliranje drugih operativnih sistema i drugih SUBP, ETL i BI alata na tim istim računarima radi izvođenja drugog eksperimenta, pa ponavljanje postupka radi izvođenja trećeg, četvrtog i narednih eksperimenata izgledalo je kao mukotrpan i nedostižan proces. Zato je odlučeno da se sve ove instalacije izvedu na virtualnim mašinama, kako bi se iste mogle kopirati na različite računare u cilju merenja performansi. Imajući u vidu da su korišćena i starija i novija rešenja koja rade na različitim operativnim sistemima (Windows desktop, Windows Server, Linux Centos 6, Centos 7 itd.) bilo je potrebno kreirati veći broj virtualnih mašina. Isto tako, u zavisnosti od instaliranog alata, na virtualnim operativnim sistemima se moralo voditi računa o instaliranim verzijama drajvera za baze podataka, kao i instaliranim klijentskim softverima za pristup bazama podataka.

Nakon instaliranja pojedinačnog SUBP na OS virtualne mašine (MS SQL na Windows Serveru, ORACLE na Centosu, MySQL na Centosu, MariaDB na Centosu, HP Vertica na Centosu itd.) bilo je potrebno učitati podatke iz OLTP i OLAP tabela u odgovarajuće baze podataka koristeći odgovarajuće dijalekte SQL-a. U OLAP okruženju učitano je oko 10-ak dimenzionih tabela veličine od 100-tinak do 60,000 redova, kao i četiri *fact* tabele veličine od 300,000 i 5,000,000 redova. Povezivanje različitih mašina je ostvareno uz pomoć VMware Network Editor-a, gde su morale da se usaglase IP adrese virtualnih mašina u odgovarajućim NAT konfiguracijama. Navedene virtualne mašine su kopirane na nekoliko računara različitih konfiguracija, kako bi se izmerio i uticaj samog hardvera na rezultate eksperimenata.

Za pojedine SUBP brzina izvršenja upita inicijalno je bila spora iz razloga što su konfiguracioni parametri imali podrazumevane vrednosti nakon instalacije. Nakon što su svi konfiguracioni fajlovi kod SUBP (koji su dozvoljavali da Administrator može da promeni navedene parametre) podešeni na optimalne vrednosti, a na osnovu nekoliko iteracija testiranja, performanse su se značajno poboljšale. Isto tako su za pojedine SUPB testirani različiti načini smeštanja i čitanja podataka (*row-store*, *column-store*, *in-memory database* itd.)

Performanse analitičkih baza podataka testirane su za on-line režim, koji podrazumeva da agregacioni podaci nisu unapred pripremani preko OLAP kocki, već se agregacija izvršavala direktno po postavljanju upita. Očekivano je dokazana pretpostavka da *column-store* platforme po performansama nadmašuju standardne baze podataka koje upisuju podatke u skladište po redovima. Rezultati su dobijeni izvršenjem upita nad SUBP u virtualnim okruženjima sa po 4GB i 8GB dodeljene memorije.

Za testiranje performansi različitih baza instaliranih na različitim operativnim sistemima korišćena su četiri kompleksna SQL agregaciona upita (Q1-Q4). Najveća brzina izvršenja eksperimenata postignuta je na najbržem računaru korišćenom u eksperimentu, na komercijalnim SUBP: HP Vertica i MS SQL Server kod kojih su tabele u bazi podataka indeksirane u *Column-store* formatu. Nekomercijalne baze podataka MariaDB ColumnStore i MySQL DB sa InfoBright mehanizmom za skladištenje koja koristi *column-store* indeksiranje su pokazale prilično dobre performanse. Oracle 12c i MariaDB CS su isto tako pokazali solidne rezultate. Generalno posmatrano, svi upiti nad *column-store* tabelama koje sadrže 5,000,000 redova su izvršavani na svim računarima za manje od 10 sekundi, dok su u eksperimentima gde je baza podataka koristila indekse *row store* u MS SQL i MySQL, ta vremena bila veća. Na osnovu izmerenih rezultata, preporuka je da se *column-store* platforme baza podataka koriste za analitičke potrebe. Imajući u vidu visok nivo integracije sa ostalim sistemima kao i odlične performanse besplatne edicije HP Vertica (HP Vertica Community Edition), poslovne organizacije bi trebalo da razmotre njeno korišćenje za potrebe izveštavanja. Ukoliko ograničenje besplatne edicije ne utiču negativno na performanse rešenja, ovaj SUBP može da predstavlja odličan izbor za mala i srednja preduzeća.

Kod poređenja performansi ETL alata dobijeni rezultati su pokazali da se najbolje performanse dobijaju kada se koristi ista platforma za bazu podataka, ETL alat i operativni sistem, jer se tada ostvaruje najveća kompatibilnost interfejsa. Interfejsi su međusobno optimizovani. Tako, na Majkrosoft platformi (kombinacija Windows Server OS – MS SQL Server – SSIS) su postignuti najbolji rezultati. Za kompletnu transformaciju podataka iz OLTP u OLAP bazu gore navedenih tabela, ETL proces je izvršen u opsegu od oko jednog minuta na najbržem računaru do oko tri minuta na najsporijem računaru. Na osnovu izmerenih rezultata, preporuka je da se za najbolje performanse izabere ETL alat kompatibilan sa bazom podataka i operativnim sistemom. Imajući u vidu mogućnost korišćenja komercijalnih i besplatnih ETL rešenja, generalna preporuka za poslovanje je da treba razmotriti korišćenje besplatnih platformi, za slučajeve kada su performanse koje te platforme nude zadovoljavajuće, što može biti čest slučaj kod malih i srednjih preduzeća.

Za potrebe testiranja performansi BI alata izabrana su dva alata: Microsoft PowerBI Desktop i Pentaho BI Server firme Hitachi Vantara koji generiše izveštaje u web okruženju. Brzina dinamičke vizualizacije izveštaja je testirana koristeći iste upite nad istom (Microsoft SQL

Server) bazom podataka. Rezultati testiranja su pokazali da su izveštaji brže generisani na desktop okruženju u Power BI alatu na Windows hostu na najbržem računaru, a da su web izveštaji brže generisani na Pentaho BI serveru na MacOS hostu. U oba slučaja brzina generisanja dinamičkih izveštaja je bila manja od 1 sekunde. Pored toga, imajući u vidu i visok nivo integracije testiranih alata sa brojnim SUBP, preporuka za poslovna preduzeća je da koriste BI rešenja koja omogućavaju distribuciju izveštaja za poslovne korisnike i koja su besplatna.

Buduće istraživanje u ispitivanju performansi SUBP, kao i ETL i BI alata bi se sastojalo u sledećem:

- Iako i komercijalni i besplatni ETL alati nude merenje vremena pojedinih koraka prilikom generisanja izveštaja, koraci u ovim alatima nisu isti, što onemogućava detaljniju analizu koji se koraci brže izvršavaju u kom alatu. Potrebno je omogućiti da se koraci u različitim alatima definišu na isti način i da se omogući merenje vremena izvršenja tih koraka, kako bi se našle konkretne razlike u performansama različitih alata.
- U nekim BI alatima je omogućeno samo merenje vremena izvršenja upita ali ne i vreme potrebno da se iz dobijenih rezultata upita vizualizuje izveštaj. Potrebno je pronaći način da se u tim alatima omogući merenje vremena koje je potrebno za vizualizaciju izveštaja.
- Zbog obima posla, testirana je kombinacija SUBP-a, ETL i BI alata samo jednog komercijalnog proizvođača – Microsoft. Potrebno je testirati i ostale kompatibilne kombinacije SUBP-a, ETL i BI alata drugih proizvođača kao što su ORACLE, IBM itd.

U skladu sa navedenim budućim planiranim istraživanjima, pratiće se i dostupna naučna literatura koja će pokazati u kojim pravcima ostali autori analiziraju ovu oblast i u skladu sa tim će se istraživanja i izvršiti.

8. Literatura

- [1] Jia-Lang Seng, S. Bing Yao, Alan R. Hevner “Requirements-driven database systems benchmark method “, *Decision Support Systems*, 38, p 629-648, 2005
- [2] S. BING YAO, ALAN R. HEVNER, HELENE YOUNG-MYERS “Analysis of Database System Architectures Using Benchmarks “, *IEEE TRANSACTIONS ON SOFTWARE ENGINEERING*, VOL. SE-13, NO. 6, pp 709-725, 1987
- [3] Philip A. Bernstein, Eric Newcomer “Principles of Transaction Processing “, Morgan Kaufmann Publishers, 2009,
- [4] Edward D. Lazowska “Quantitative System Performance, Computer System Analysis Using Queuing Network Models “, Prentice Hall, Inc. 1984
- [5] Rasha Osman, William J. Knottenbelt “Database system performance evaluation models: A survey “, *Performance Evaluation* 69, pp 471–493, 2012
- [6] Youssef Bassil “A Comparative Study on the Performance of the Top DBMS Systems “, *Journal of Computer Science & Research (JCSCR)* - ISSN 2227-328X, Vol. 1, No. 1, Pages. 20-31, 2012
- [7] Udai Shanker, Manoj Misra, Anil K. Sarje “Distributed real time database systems: background and literature review “, *Distrib Parallel Databases* 23, pp 127–149, 2008
- [8] Amlanjyoti Saikia, Sherin Joy, Dhondup Dolma, Roseline Mary “Comparative Performance Analysis of MySQL and SQL Server Relational Database Management Systems in Windows Environment “, *International Journal of Advanced Research in Computer and Communication Engineering* Vol. 4, Issue 3, pp 160-164, March 2015
- [9] Khawar Islam, Kamran Ahsan, S.A.K. Bari, Muhammad Saeed, Syed Asim Ali “Huge and Real-Time Database Systems: A Comparative Study and Review for SQL Server 2016, Oracle 12c & MySQL 5.7 for Personal Computer “, *Journal of Basic & Applied Sciences*, 3, pp 481-490, 2017
- [10] Cătălin Tudose, Carmen Odubășteanu “Database Management Systems Comparative Study: Performances of Microsoft SQL Server Versus Oracle”, 3rd Romanian-Hungarian Joint Symposium on Applied Computational Intelligence (SACI), May 25-26, 2006, Timisoara, Romania
- [11] Hristo Kyurkchiev, Kalinka Kaloyanova “Performance Study of Analytical Queries of Oracle and Vertica “, *Conference: Information Systems & Grid Technologies Sofia*, DOI:10.13140/2.1.3667.0726, 2013

- [12] Rotsnarani Sethy, Santosh Kumar Dash, Mrutyunjaya Panda “Performance Comparison Between Apache Hive and Oracle SQL for Big Data Analytics “, Proceedings of the Eighth International Conference on Soft Computing and Pattern Recognition (SoCPaR 2016) pp 130–141, Vellore, India, December 19–21, 2016
- [13] Sudhanshu Kulshrestha, Shelly Sachdeva “Performance Comparison for Data Storage - Db4o and MySQL Databases”, IEEE, 2014, DOI: 10.1109/IC3.2014.6897167
- [14] Vassilis Kilintzis, Nikolaos Beredimas, Ioanna Chouvarda “Evaluation of the performance of open-source RDBMS and triplestores for storing medical data over a web service”, IEEE, 2014, DOI: 10.1109/EMBC.2014.6944623
- [15] Amit Kumar Dwivedi, C.S. Lamba, Shweta Shukla “Performance Analysis of Column Oriented Database versus Row Oriented Database”, International Journal of Computer Applications (0975 – 8887) Volume 50 – No.14, July 2012, pp 31- 34
- [16] Punam Bajaj Simranjit Kaur Dhindsa “A Comparative Study of Database Systems”, International Journal of Engineering and Innovative Technology Volume 1, Issue 6, June 2012, pp 267-269
- [17] Daniel J. Abadi, Samuel R. Madden, Nabil Hachem “ColumnStores vs. RowStores: How Different Are They Really?”, SIGMOD '08: Proceedings of the 2008 ACM SIGMOD international conference on Management of data, June 2008, Vancouver, BC, Canada, Pages 967–980
- [18] Daniel J. Abadi “Column-Stores for Wide and Sparse Data”, Conference: CIDR 2007, Third Biennial Conference on Innovative Data Systems Research, Asilomar, CA, USA, 2007
- [19] Anastassia Ailamaki, David J. DeWitt, Mark D. Hill, Marios Skounakis “Weaving Relations for Cache Performance”, VLDB '01: Proceedings of the 27th International Conference on Very Large Data Bases, September 2001, Roma, Italy, Pages 169–180
- [20] Siniša Ilić, Dragiša Miljković, Slobodan Obradović, Alempije Veljović, Bojan Jovanović “COMPARISON OF PERFORMANCE IN DATA ANALYSIS IN DEDICATED AND TRADITIONAL DBMS”, INTERNATIONAL SCIENTIFIC CONFERENCE, 18 – 19 November 2016, GABROVO, Bulgaria, pp II-219 - 225
- [21] Tilmann Rabl, Meikel Poess “Parallel Data Generation for Performance Analysis of Large, Complex RDBMS”, Conference: Proceedings of the Fourth International Workshop on Testing Database Systems, DBTest 2011, Athens, Greece, June 13, 2011, DOI:10.1145/1988842.1988847

- [22] Swati Jain, Paras Nath Barwal “Performance Analysis of Optimization Techniques for SQL Multi Query Expressions Over Text Databases in RDBMS”, International Journal of Information & Computation Technology. ISSN 0974-2239 Volume 4, Number 8 (2014), pp. 841-852
- [23] G. Lawrence Sanders, Seungkyoon Shin “Denormalization Effects on Performance of RDBMS”, IEEE, 2001, DOI: 10.1109/HICSS.2001.926306
- [24] Ayub M. B., Ali N. “Performance comparison of in-memory and disk-based databases using transaction processing performance council (TPC) benchmarking”, Academic Journals, Vol. 8(1), pp. 1-8, August 2018, DOI 10.5897/JIIS2018.0106
- [25] Sasalak Tongkaw, Aumnat Tongkaw “A Comparison of Database Performance of MariaDB and MySQL with OLTP Workload”, IEEE, 2016, DOI: 10.1109/ICOS.2016.7881999
- [26] Jing Han, Haihong E, Guan Le, Jian Du “Survey on NoSQL Database”, IEEE, 2011, DOI: 10.1109/ICPCA.2011.6106531
- [27] Miral Borad, Tejash Chauhan, Harsh Meht “A Review on NoSQL Databases”, International Journal of Science and Research (IJSR), Paper ID: 6(11), ART20177781, pp 26-28, 2015
- [28] Md. Razu Ahmed, Mst. Arifa Khatun, Md. Asraf Ali, Kenneth Sundaraj “A literature review on NoSQL database for big data processing”, International Journal of Engineering & Technology, 7 (2), pp 902-906, 2018
- [29] Bogdan George Tudorica, Cristian Bucur “A comparison between several NoSQL databases with comments and notes”, IEEE, 2011, DOI: 10.1109/RoEduNet.2011.5993686
- [30] Khalid Mahmood “Performance Comparison of NOSQL Database Cassandra and SQL Server for Large Databases”, Journal of Independent Studies and Research – Computing Volume 14 Issue 2, pp 21-25, 2016
- [31] Cornelia Györödi, Robert Györödi, Alexandra Ștefan, Livia Bandici, “A Comparative Study of Databases with Different Methods of Internal Data Management”, (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 7, No. 4, pp 266-271, 2016
- [32] Azhi Faraj, Bilal Rashid, Twana Shareef “COMPARATIVE STUDY OF RELATIONAL AND NONRELATIONS DATABASE PERFORMANCES USING ORACLE AND MONGODB SYSTEMS”, INTERNATIONAL JOURNAL OF COMPUTER ENGINEERING & TECHNOLOGY (IJCET), Volume 5, Issue 11, pp. 11-22, 2014

- [33] Alexandru Boicea, Florin Radulescu, Laura Ioana Agapin “MongoDB vs Oracle - database comparison”, 2012 Third International Conference on Emerging Intelligent Data and Web Technologies, IEEE, 2012, DOI 10.1109/EIDWT.2012.32
- [34] Houcine Matallah, Ghalem Belalem, Algeria Karim Bouamrane “Comparative Study Between the MySQL Relational Database and the MongoDB NoSQL Database”, International Journal of Software Science and Computational Intelligence, 2021, DOI: 10.4018/IJSSCI.2021070104
- [35] Dipina Damodaran B, Shirin Salim, Surekha Mariam Vargese “MONGODB VS MYSQL: A COMPARATIVE STUDY OF PERFORMANCE IN SUPER MARKET MANAGEMENT SYSTEM”, International Journal of Computational Science and Information Technology (IJCSITY) Vol.4, No.2, pp 31- 38, 2016
- [36] Sanobar Khan, Prof.Vanita Mane “SQL Support over MongoDB using Metadata”, International Journal of Scientific and Research Publications, Volume 3, Issue 10, 2013, ISSN 2250-3153
- [37] Sameera Abbas FADHEL, Enas Ali JAMEEL “A COMPARISON BETWEEN NOSQL AND RDBMS: STORAGE AND RETRIEVAL”, International Journal of Applied Sciences and Technology, Volume 4, Issue 3, pp 172- 184, 2022
- [38] Muhammad Zohaib Khan, Fahim Uz Zaman, Muhammad Adnan, Aisha Imroz, Mahira Abdul Rauf, Zuhaib Phul “Comparative Case Study: An Evaluation of Performance Computation Between SQL And NoSQL Database”, Sindh Journal of Headways in Software Engineering, Vol 1 No 2 pp 14-23, 2023
- [39] Rachit Pandey “Performance Benchmarking and Comparison of Cloud-Based Databases MongoDB (NoSQL) Vs MySQL (Relational) using YCSB”, 2020, DOI: 10.13140/RG.2.2.10789.32484
- [40] Mit Jain, Ashish Khanchandani, Cajetan Rodrigues “Performance Comparison of Graph Database and Relational Database”, May 2023, DOI: 10.13140/RG.2.2.27380.32641
- [41] Cornelia A. Gyorödi, Diana V. Dumse-Burescu, Doina R. Zmaranda, Robert S. Gyorödi , Gianina A. Gabor, George D. Pecherle “Performance Analysis of NoSQL and Relational Databases with CouchDB and MySQL for Application’s Data Storage”, Applied Sciences Volume 10 Issue 23, 2020, doi:10.3390/app10238524
- [42] Uwe Hohenstein and Martin Jergler “Database Performance Comparisons: An Inspection of Fairness”, Proceedings of the 8th International Conference on Data Science, Technology and Applications DATA - Volume 1, 243-250, 2019, DOI: 10.5220/0007926602430250

- [43] Lucas Ferreira da Silva, João V. F. Lima “An evaluation of relational and NoSQL distributed databases on a low-power cluster”, *The Journal of Supercomputing* 79, pp 13402–13420, 2023
- [44] Yishan Li and Sathiamoorthy Manoharan “A performance comparison of SQL and NoSQL databases”, *IEEE*, 2013, DOI: 10.1109/PACRIM.2013.6625441
- [45] Monika Sharma, Vishal Deep Sharma, Mahesh M. Bundeale “Performance Analysis of RDBMS and No SQL Databases: PostgreSQL, MongoDB and Neo4j”, *IEEE*, 2018, DOI: 10.1109/ICRAIE.2018.8710439
- [46] A B M Moniruzzaman “NewSQL: Towards Next-Generation Scalable RDBMS for Online Transaction Processing (OLTP) for Big Data Management”, *International Journal of Database Theory and Application* Vol.7, No.6, pp.121-130, 2014, DOI:10.14257/ijda.2014.7.6.11
- [47] J. Bryan Osborne “Performance Impact of Type-I Virtualization on a NewSQL Relational Database Management System”, (*IJACSA*) *International Journal of Advanced Computer Science and Applications*, Vol. 13, No. 1, 2022, DOI: 10.14569/IJACSA.2022.0130101
- [48] Zhen Hua Liu, Beda Hammerschmidt, Doug McMahon, Ying Lu, Hui Joe Chang “Closing the functional and Performance Gap between SQL and NoSQL”, *SIGMOD '16: Proceedings of the 2016 International Conference on Management of Data*, Pages 227–238, 2016, DOI:10.1145/2882903.2903731
- [49] Rafael Almeida, Jorge Bernardino “Testing SQL and NoSQL approaches for big data warehouse systems”, *Int. J. Business Process Integration and Management*, Vol. 7, No. 4, pp 322-334, 2015, DOI:10.1504/IJBPIIM.2015.073656
- [50] Douglas Kunda, Hazael Phiri “A Comparative Study of NoSQL and Relational Database”, *Zambia ICT Journal* 1(1): pp 1-4, 2017, <https://doi.org/10.33260/zictjournal.v1i1.8>
- [51] Twana Hussein Shareef, Karzan Hussein Shareef, Bilal Najmaddin Rashid “A Survey of Comparing Different Cloud Database Performance: SQL and NoSQL”, *Passer Journal*, Passer 4, pp 45- 57, 2022, DOI:10.24271/psr.2022.301247.1104
- [52] Toni Taipalus, “Database management system performance comparisons: A systematic literature review”, October 2023, *Journal of Systems and Software* 208(2):111872, DOI:10.1016/j.jss.2023.111872
- [53] Wisal Khan, Teerath Kumar, Cheng Zhang, Kislay Raj, Arunabha M. Roy, Bin Luo “SQL and NoSQL Database Software Architecture Performance Analysis and Assessments—A Systematic Literature Review”, *Big Data Cogn. Comput.* 2023, 7(2), 97; <https://doi.org/10.3390/bdcc7020097>

- [54] Abdullah Talha Kabakus, Resul Kara “A performance evaluation of in-memory databases”, Journal of King Saud University – Computer and Information Sciences (2017) 29, pp 520–525, DOI:10.1016/j.jksuci.2016.06.007
- [55] Robin Hecht, Stefan Jablonski “NoSQL evaluation: A use case oriented survey”, 2011 International Conference on Cloud and Service Computing, DOI: 10.1109/CSC.2011.6138544
- [56] Cristina BĂZĂR, Cosmin Sebastian IOSIF “The Transition from RDBMS to NoSQL. A Comparative Analysis of Three Popular Non-Relational Solutions: Cassandra, MongoDB and Couchbase”, Database Systems Journal, Academy of Economic Studies - Bucharest, Romania, vol. 5(2), pages 49-59, September.
- [57] EnqingTang, Yushun Fan “Performance Comparison between Five NoSQL Databases”, IEEE, 7th International Conference on Cloud Computing and Big Data (CCBD), 2016, DOI: 10.1109/CCBD.2016.030
- [58] Diogo Augusto Pereira, Wagner Ourique de Morais & Edison Pignaton de Freitas “NoSQL real-time database performance comparison”, International Journal of Parallel, Emergent and Distributed Systems, 2017, DOI: 10.1080/17445760.2017.1307367
- [59] Mrs. Rupali M. Chopade, Mr. Nikhil S. Dhavase “MongoDB, CouchBase: Performance Comparison for Image Dataset”, IEE, 2nd International Conference for Convergence in Technology (I2CT), 2017, pp 255-258, DOI: 10.1109/I2CT.2017.8226131
- [60] Gerard Haughian, Rasha Osman, William J. Knottenbelt, Benchmarking Replication in Cassandra and MongoDB NoSQL Datastores, International Conference on Database and Expert Systems Applications, 2016, DOI:10.1007/978-3-319-44406-2_12
- [61] A. GORBENKO, A. KARPENKO, O. TARASYUK “PERFORMANCE EVALUATION OF VARIOUS DEPLOYMENT SCENARIOS OF THE 3-REPLICATED CASSANDRA NOSQL CLUSTER ON AWS”, Radioelectronic and Computer Systems (4). pp. 157-165. ISSN 2663-2012 DOI: <https://doi.org/10.32620/reks.2021.4.13>
- [62] Anatoliy Gorbenko, Alexander Romanovsky, Olga Tarasyuk “Interplaying Cassandra NoSQL Consistency and Performance: a Benchmarking Approach”, Dependable Computing - EDCC 2020 Workshops (pp.168-184), 2020, DOI:10.1007/978-3-030-58462-7_14
- [63] A. GORBENKO, O. TARASYUK “EXPLORING TIMEOUT AS A PERFORMANCE AND AVAILABILITY FACTOR OF DISTRIBUTED REPLICATED DATABASE SYSTEMS”, RADIOELECTRONIC AND COMPUTER SYSTEMS (4). pp. 98-105, 2020, ISSN 1814-4225 DOI: <https://doi.org/10.32620/reks.2020.4.09>

- [64] Mohammad Abu Kausar, Mohammad Nasar, Arockiasamy Soosaimanickam “A Study of Performance and Comparison of NoSQL Databases: MongoDB, Cassandra, and Redis Using YCSB”, *INDIAN JOURNAL OF SCIENCE AND TECHNOLOGY*, 2020, pp 1532-1540, DOI: 10.17485/IJST/v15i31.1352
- [65] Ali Al-Allawee, Pascal Lorenz, Abdelhafid Abouaissa, Mosleh Abualhaj “A Performance Evaluation of In-Memory Databases Operations in Session Initiation Protocol”, *Network* 2023, 3(1), 1-14; <https://doi.org/10.3390/network3010001>
- [66] Francini Hak, Tiago Guimarães, António Abelha, Manuel Santos “An Exploratory Study of a NoSQL Database for a Clinical Data Repository”, *WorldCIST 2020: Trends and Innovations in Information Systems and Technologies* pp 476–483, DOI:10.1007/978-3-030-45697-9_46
- [67] Prashanth Jakkula “HBase or Cassandra? A Comparative study of NoSQL Database Performance”, *International Journal of Scientific and Research Publications*, Volume 10, Issue 3, March 2020, pp 808-820, DOI:10.29322/IJSRP.10.03.2020.p9999
- [68] Cornelia A. Gyorödi, Diana V. Dum,se-Burescu, Doina R. Zmaranda and Robert, S. Gyorödi “A Comparative Study of MongoDB and Document-Based MySQL for Big Data Application Data Management”, *Big Data Cogn. Comput.* 2022, 6(2), 49; <https://doi.org/10.3390/bdcc6020049>
- [69] Rupali Kaur, Jaspreet Kaur Sahiwal “A review of comparison between NoSQL Databases: MongoDB and CouchDB”, *International Journal of Recent Technology and Engineering (IJRTE)*, ISSN: 2277-3878, Volume-7, Issue-6S, March 2019, pp 892-898
- [70] Houcine Matallah, Ghalem Belalem “Evaluation of NoSQL Databases: MongoDB, Cassandra, HBase, Redis, Couchbase, OrientDB”, *International Journal of Software Science and Computational Intelligence*, Volume 12, Issue 4, 2020, pp 71- 91, DOI:10.4018/IJSSCI.2020100105
- [71] Sachin Chaudhary, Devendra Prasad Murala, V. K. Srivastav “A Critical Review of Data Warehouse”, *Global Journal of Business Management and Information Technology*, Volume 1, Number 2 (2011), pp. 95-103, <https://www.ripublication.com/Volume/gjbmit1n2.htm>
- [72] Mr. Dishek Mankad, Mr. Preyash Dholakia “The Study on Data Warehouse Design and Usage”, *International Journal of Scientific and Research Publications*, Volume 3, Issue 3, March 2013, ISSN 2250-3153, <https://www.ijsrp.org/research-paper-0313/ijsrp-p1597.pdf>
- [73] Mutaz M. Al-Debei “Data Warehouse as a Backbone for Business Intelligence: Issues and Challenges”, *European Journal of Economics, Finance and Administrative Sciences*, ISSN 1450-2275 Issue 33 (2011), pp153-166

- [74] Saeed Rouhani, Sara Asgari, Seyed Vahid Mirhosseini “Review Study: Business Intelligence Concepts and Approaches”, American Journal of Scientific Research, ISSN 1450-223X Issue 50 (2012), pp. 62-75
- [75] OYKU ISIK, MARY C. JONES, ANNA SIDOROVA “Business intelligence (BI) success and the role of BI capabilities”, INTELLIGENT SYSTEMS IN ACCOUNTING, FINANCE AND MANAGEMENT, Intell. Sys. Acc. Fin. Mgmt. 18, 161–176 (2011), DOI:10.1002/isaf.329
- [76] Panos Vassiliadis “A Survey of Extract–Transform–Load Technology”, September 2011, International Journal of Data Warehousing and Mining 5(3):1-27, DOI:10.4018/jdwm.2009070101
- [77] Nitin Anand “Application of ETL Tools in Business Intelligence”, International Journal of Scientific and Research Publications, Volume 2, Issue 11, November 2012, ISSN 2250-3153, <https://www.ijsrp.org/research-paper-1112.php?rp=P11356>
- [78] Jose Barateiro, Helena Galhardas “A Survey of Data Quality Tools”, Datenbank-Spektrum 14(5):15-21, 2005, https://www.researchgate.net/publication/220102796_A_Survey_of_Data_Quality_Tools
- [79] Manel Souibguia,b, Faten Atiguib, Saloua Zammalia, Samira Cherfib, Sadok Ben Yahiaa “Data quality in ETL process: A preliminary study”, 23rd International Conference on Knowledge-Based and Intelligent Information & Engineering Systems, ScienceDirect, Procedia Computer Science 159 (2019) pp 676–687
- [80] J P A Runtuwene, I R H T Tangkawarow, C T M Manoppo, R J Salaki “A Comparative Analysis of Extract, Transformation and Loading (ETL) Process”, IOP Conf. Series: Materials Science and Engineering 306 (2018) 012066 doi:10.1088/1757-899X/306/1/012066
- [81] Abdellah Amine, Rachid Ait Daoud, Belaid Bouikhalene “Efficiency Comparaison and Evaluation between Two ETL Extraction Tools”, Indonesian Journal of Electrical Engineering and Computer Science, Vol. 3, No. 1, July 2016, pp. 174 ~ 181, DOI:10.11591/ijeecs.v3.i1.pp174-181
- [82] Sinisa S. Ilic, Dragisa Miljkovic, Alempije Veljovic, Jasmina Novakovic, Vladimir Veljovic “COMPARISON OF PERFORMANCES OF BUILT-IN TOOLS FOR LOAD AND EXPORT OF DATA TO AND FROM POPULAR DATABASES”, COMPUTER SCIENCE AND TECHNOLOGIES, 2018, Number 1/2018 pp 32-38
- [83] Siniša S. Ilić, Danijela Milentijević, Alempije Veljović, Slobodan Obradović, Aleksandar Žorić, Nebojša Arsić “THE IMPLEMENTATION OF ETL PROCESSES ON EXAMPLE OF

AUTHORISED DLS”, INTERNATIONAL SCIENTIFIC CONFERENCE “UNITECH 2015”, GABROVO, Bulgaria, pp II-183 – II-188

[84] Rajendrani Mukherjee, Pragma Kar “A Comparative Review Of Data Warehousing ETL Tools With New Trends And Industry Insight”, 2017 IEEE 7th International Advance Computing Conference, pp 943-948, DOI: 10.1109/IACC.2017.0192

[85] Amanpartap Singh PALL, Dr. Jaiteg Singh KHAIRA “A comparative Review of Extraction, Transformation and Loading Tools”, Database Systems Journal, Academy of Economic Studies - Bucharest, Romania, vol. 4(2), pages 42-51, July. <https://ideas.repec.org/a/aes/dbjour/v4y2013i2p42-51.html>

[86] Rahmadi Wijaya, S.Si., MT, Bambang Pudjoatmodjo, S.Si., MT. “An Overview and Implementation of Extraction-Transformation-Loading (ETL) Process In Data Warehouse (Case Study: Department of Agriculture)”, 2015 3rd International Conference on Information and Communication Technology (ICoICT), 2015, DOI:10.1109/ICoICT.2015.7231399

[87] Asma Qaiser, Muhammad Umer Farooq, Syed Muhammad Nabeel Mustafa, Nazia Abrar, Comparative Analysis of ETL Tools in Big Data Analytics, Pakistan Journal of Engineering and Technology, PakJET, ISSN (e): 2664-2050, Volume: 6, Number: 1, Pages: 7- 12, Year: 2023, https://www.researchgate.net/publication/369094822_Comparative_Analysis_of_ETL_Tools_in_Big_Data_Analytics

[88] Adnan, Amil Ahmad Ilham, Syahrul Usman “Performance Analysis of Extract, Transform, Load (ETL) in Apache Hadoop atop NAS Storage using ISCSI”, IEEE, 2017 4th International Conference on Computer Applications and Information Processing Technology (CAIPT), 2017, DOI: 10.1109/CAIPT.2017.8320716

[89] Eckerson, W., White, C.: Evaluating ETL and Data Integration Platforms. THE DATA WAREHOUSING INSTITUTE, A 101Communications Publication, http://download.101com.com/tdwi/research_report/2003ETLReport.pdf

[90] Sumit Misra, Sanjoy Kumar Saha, and Chandan Mazumdar “Performance Comparison of Hadoop Based Tools with Commercial ETL Tools – A Case Study”, BDA 2013, LNCS 8302, pp. 176–184, 2013, DOI:10.1007/978-3-319-03689-2_12

[91] Sonali Vyas, Pragya Vaishnav “A comparative study of various ETL process and their testing techniques in data warehouse”, Journal of Statistics and Management Systems, 20:4, 753-763, DOI: 10.1080/09720510.2017.1395194, 2017

- [92] Seema Maitreya, C.K. Jha “MapReduce: Simplified Data Analysis of Big Data”, ScienceDirect, Procedia Computer Science 57 (2015) 563 – 571, doi: 10.1016/j.procs.2015.07.392
- [93] Junghee Jo, Kang-Woo Lee “MapReduce-Based D_ETL Framework to Address the Challenges of Geospatial Big Data”, International Journal of Geo-Information, 2019, 8, 475; doi:10.3390/ijgi8110475
- [94] Papa Senghane Diouf, Aliou Boly, Samba Ndiaye “Performance of the ETL processes in terms of volume and velocity in the cloud: state of the art”, IEEE, 2017 4th IEEE International Conference on Engineering Technologies and Applied Sciences (ICETAS), 2017, DOI: 10.1109/ICETAS.2017.8277875
- [95] Dhamotharan Seenivasan “Improving the Performance of the ETL Jobs”, International Journal of Computer Trends and Technology Volume 71 Issue 3, 27-33, March 2023, DOI:10.14445/22312803/IJCTT-V71I3P105
- [96] Vishal Gour, Dr. S.S.Sarangdevot, Govind Singh Tanwar, Anand Sharma “Improve Performance of Extract, Transform and Load (ETL) in Data Warehouse”, (IJCSSE) International Journal on Computer Science and Engineering, Vol. 02, No. 03, 2010, 786-789, https://www.researchgate.net/publication/49618608_Improve_Performance_of_Extract_Transform_and_Load_ETL_in_Data_Warehouse
- [97] Suyeon Lee, Sungyong Park “Performance Analysis of Big Data ETL Process over CPU-GPU Heterogeneous Architectures”, 2021 IEEE 37th International Conference on Data Engineering Workshops (ICDEW), pp 42- 47, DOI: 10.1109/ICDEW53142.2021.00015
- [98] Shaker H. Ali El-Sappagh a, Abdeltawab M. Ahmed Hendawi, Ali Hamed El Bastawissy “A proposed model for data warehouse ETL processes”, Journal of King Saud University – Computer and Information Sciences (2011) 23, 91–104, DOI:10.1016/j.jksuci.2011.05.005
- [99] Asma Dhaouadi, Khadija Bousselmi, Mohamed Mohsen Gammoudi, Sébastien Monnet, Slimane Hammoudi “DataWarehousing Process Modeling from Classical Approaches to New Trends: Main Features and Comparisons”, Data 2022, 7(8), 113; <https://doi.org/10.3390/data7080113>
- [100] Dhamotharan Seenivasan “ETL (Extract, Transform, Load) Best Practices”, International Journal of Computer Trends and Technology Volume 71 Issue 1, 40-44, January 2023, DOI:10.14445/22312803/IJCTT-V71I1P106
- [101] Višnja Istrat, Sanja Stanisavljev, Branko Markoski “THE ROLE OF BUSINESS INTELLIGENCE IN DECISION PROCESS MODELING “, The European Journal of Applied Economics, 12(2), 44-52, 2015

- [102] Bernhard Wieder, Maria-Luise Ossimitz “The impact of Business Intelligence on the quality of decision making – a mediation model“, Conference on ENTERprise Information Systems / CENTERIS 2015 October 7-9, 2015, Procedia Computer Science 64 (2015) 1163 – 1171
- [103] Saeed Rouhani Amir Ashrafi Ahad Zare Ravasan Samira Afshari “The impact model of business intelligence on decision support and organizational benefits “, Journal of Enterprise Information Management, Vol. 29 Iss 1 pp. 19 – 50, <http://dx.doi.org/10.1108/JEIM-12-2014-0126>
- [104] Jakub Andar, Petra Kasparova “Impact of Management Support on Business Intelligence Adoption: Illustrative Case Study Testing Different Managerial Strategies“, Acta Informatica Pragensia 2024, Volume 13, Issue 1, pp. x–x, <https://doi.org/10.18267/j.aip.230>
- [105] Mariem Belghitha, Hanen Ben Ammarb, Abdelkarim Elloumia, Wafik Hachichac “A new rolling forecasting framework using Microsoft Power BI for data visualization: A case study in a pharmaceutical industry “, Annales Pharmaceutiques Françaises, Volume 82, Issue 3, May 2024, Pages 493-506, <https://doi.org/10.1016/j.pharma.2023.10.013>
- [106] Shenwei Zhang “Analysis of Business Intelligence Technology in the Big Data Era - A Case Study of Alibaba Group“, Proceedings of the 3rd International Conference on Business and Policy Studies, DOI: 10.54254/2754-1169/74/20241520
- [107] Mohammad Taghi Sadeghi, Ibaa Al hasan “Enhancing Supply Chain Management in Small-Medium Enterprises through Business Intelligence: A Conceptual Model Approach “, Research Square, <https://doi.org/10.21203/rs.3.rs-4249032/v1>
- [108] Romeu Sequeira, Arsénio Reis, Paulo Alves and Frederico Branco “Roadmap for Implementing Business Intelligence Systems in Higher Education Institutions: Systematic Literature Review“, 20th International Conference on Smart Business Technologies (ICSBT 2023). Rome, Italy, 11–13 July 2023, Information 2024, 15, 208, <https://doi.org/10.3390/info15040208>
- [109] Radhakrishnan Subramaniam, Prashobhan Palakeel, Manimuthu Arunmozhi, Manikandan Sridharan, Uthayakumar Marimuthu “Factors driving business intelligence adoption: an extended technology-organization-environment framework “, Indonesian Journal of Electrical Engineering and Computer Science, Vol. 34, No. 3, June 2024, pp. 1893~1903, ISSN: 2502-4752, DOI: 10.11591/ijeecs.v34.i3.pp1893-1903
- [110] Ashraf Bany Mohammed, Manaf Al-Okaily, Dhia Qasim, Mohammad Khalaf Al-Majali “Towards an understanding of business intelligence and analytics usage: Evidence from the banking industry “, International Journal of Information Management Data Insights 4 (2024) 100215, <https://doi.org/10.1016/j.ijime.2024.100215>

- [111] Adetumi Adewumi, Ese Eigbadon Oshioste, Onyeka Franca Asuzu, Ndubuisi Leonard Ndubuisi, Kehinde Feranmi Awonnuga and Onyeka Henry Daraojimba “Business intelligence tools in finance: A review of trends in the USA and Africa “, World Journal of Advanced Research and Reviews, 2024, 21(03), 608–616, <https://doi.org/10.30574/wjarr.2024.21.3.0333>
- [112] Sadi Badi “Enhancing Business Intelligence: Harnessing Text Analytics, AI, and ERP for Data-Driven Decision Making “, March 2024, DOI:10.13140/RG.2.2.31479.43681
- [113] Asad Abbas “Transforming Employee Performance Prediction: Harnessing Machine Learning and Analytics for Business Intelligence Advancement “, March 2024, DOI: 10.13140/RG.2.2.10770.06087
- [114] Md. Ashrafuzzaman, Israt Sultana, Rabita Rahman Era, Arjun Thimmaiah, Amit Kumar Sharma, Saira Binte Awal, Jitendra Sharma “Transforming Medical Tourism Through an Integrated Business Intelligence Platform: Data-Driven Insights and Healthcare Industry Impact “, February 2024, IGI Global, DOI: 10.4018/979-8-3693-1210-0.ch010
- [115] Malay sarkar, Rasel Mahmud Jewel, Md Salim Chowdhury, Md Al-Imran, Rumana Shahid, Aisharya Roy Puja, Rejon Kumar Ray and Sandip Kumar Ghosh “Revolutionizing Organizational Decision-Making for Stock Market: A Machine Learning Approach with CNNs in Business Intelligence and Management “, Journal of Business and Management Studies, PUBLISHED: 13 February 2024, DOI: 10.32996/jbms.2024.6.1.16
- [116] Murat Ozdemir, Eyup Emre Ulku and Kazim Yildiz “Analysis and Comparison of Business Intelligence Tools Most Preferred by Companies in Turkey “, BALKAN JOURNAL OF ELECTRICAL & COMPUTER ENGINEERING, Vol. 11, No. 2, April 2023, p.144-155
- [117] Carlo DELL’AQUILA, Francesco DI TRIA, Ezio LEFONS, and Filippo TANGORRA “Business Intelligence Systems: a Comparative Analysis“, WSEAS TRANSACTIONS on INFORMATION SCIENCE & APPLICATIONS, ISSN: 1790-0832, Issue 5, Volume 5, 2008, p.612-621
- [118] Vipin Kumar Saraogi, Jitender Kumar, Dr. A.K. Malik “DATA ANALYTICS BUSINESS INTELLIGENCE TOOLS: A COMPARATIVE STUDY“, IJESR/NCSET-2020/ Conf. Issue/ Article No-1/1-3, ISSN 2277-2685
- [119] Fehmi Skender, Violeta Manevska, “Data Visualization Tools - Preview and Comparison “, Preview and Comparison. Journal of Emerging Computer Technologies, 2(1), 30-35. Volume:2, No:1, Year: 2022, Pages: 30-35, June 2022, Journal of Emerging Computer Technologies
- [120] Danella Patrick; Abdullah Ayub Khan; Fayyaz Ali; Irfan Ali Kandhro; Mahnoor Anwar; Asadullah Kehar; Anwar Ali Sanjrani, “A novel comparison of data analytics and business

intelligence tools: an information preservation and ledger management solution “, International Journal of Electronic Security and Digital Forensics, 2023 Vol.15 No.4, pp.387 – 412, DOI: 10.1504/IJESDF.2023.131957

[121] Victor M. Parra, Azeem Mohammad, Ali Syed, Malka N. Halgamuge “Pentaho and Jaspersoft: A Comparative Study of Business Intelligence Open Source Tools Processing Big Data to Evaluate Performances “, (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 7, No. 10, 2016, p.20-29

[122] Andreia Brandão, Eliana Pereira, Marisa Esteves, Filipe Portela, Manuel Filipe Santos, António Abelha and José Machado “A Benchmarking Analysis of Open-Source Business Intelligence Tools in Healthcare Environments“, Information 2016, 7, 57; doi:10.3390/info7040057

Spisak tabela

Tabela 1. Benčmark test parametri za računare korišćene u eksperimentima	55
Tabela 2. Vreme (u ms) potrebno za izvršavanje upita za različite „starije“ SUBP na različitim računarima i različitoj memoriji dodeljenoj u Eksperimentu 1.	65
Tabela 3. Vreme (u ms) potrebno za izvršavanje upita za različite „novije“ SUBP na različitim računarima i različitoj memoriji dodeljenoj u eksperimentu 2.....	65
Tabela 4. Vreme izvršenja ETL poslova u sekundama	85
Tabela 5. Rezultati eksperimenata BI alata prikazani u milisekundama	101

Spisak slika

Slika 1. Deo šeme VMart OLTP baze	73
Slika 2. VMart OLTP tabele dobijene transformacijom OLTP tabela sa Slike 1	75
Slika 3. OLE DB Source i OLE DB Destination komponente	76
Slika 4. OLE DB Source editor	76
Slika 5. Mapiranje kolona za tabelu public_call_center_dimension	77
Slika 6. Primer SSIS posla u Data Flow Editoru	78
Slika 7. Sort Transformation Editor	79
Slika 8. Merge Join Transformation Editor	79
Slika 9. Control Flow Editor za ceo ETL posao	80
Slika 10. Table input Editor	81
Slika 11. Sort rows Editor	82
Slika 12. Merge join Editor	82
Slika 13. Editor mapiranja kolona podataka nakon transformacija i destinacione tabele	83
Slika 14. Primer PDI transformacije 5 OLTP u jednu OLAP tabelu	83
Slika 15. Izvršenje ETL posla transformacija OLTP tabela u listu Dimension i Fact OLAP tabela	84
Slika 16. Navigator prozor za izbor tabela iz baze podataka	94
Slika 17. Kreiranje relacija između tabela u prozoru Create Relationship	95
Slika 18. Izbor kolona izveštaja	96
Slika 19. Pražnjenje keša iz Data Cache management opcija	97
Slika 20. Početna web strana Pentaho Business Analytics	98
Slika 21. Pentaho Metadata Editor za definisanje Metadata Source u Pentaho BI	98
Slika 22. Kreiranje relacija između tabela u prozoru Create Relationship	99
Slika 23. Upis upita u polju Query u Datasource Panelu	99
Slika 24. JavaScript koji prikazuje sistemsko vreme započetog procesa	100
Slika 25. JavaScript koji prikazuje sistemsko vreme završetka izveštaja	100

Spisak grafikona

Grafikon 1. Brzina izvršavanja upita Q1	66
Grafikon 2. Brzina izvršavanja upita Q1 i Q3 sa većim Fact tabelama na najbržem računaru koji se koristi u eksperimentima	67
Grafikon 3. Brzina izvršavanja upita Q2 i Q4 sa manjim Fact tabelama na najbržem računaru koji se koristi u eksperimentima	68
Grafikon 4. ukupna vremena izvršenja svih ETL poslova za svaki od alata i za sve tri različite hardverske platforme u sekundama	86
Grafikon 5. Rezultati eksperimenata BI alata prikazani u milisekundama.....	102