

Оптимизација претраге текстуалних дигиталних докумената***An Optimization of Textual Digital Document Search***Наталија Шашић, *Факултет техничких наука, Нови Сад***Студијски програм – СОФТВЕРСКО
ИНЖЕЊЕРСТВО И ИНФОРМАЦИОНЕ
ТЕХНОЛОГИЈЕ**

Кратак садржај – Рад истражује употребу *Elasticsearch*-а за оптимизацију претраге у реалном времену на великим скуповима података, са посебним фокусом на научне радове из *ArXiv*-а. Истиче напредне функције *Elasticsearch*-а као што су *fuzzy* претрага, обрада синонима и скалабилност, које побољшавају ефикасност претраживања и корисничко искуство. У раду се такође разматрају изазови у оптимизацији перформанси претраге и конфигурација, нудећи увид у ефикасне стратегије за управљање великим подацима. Посебна пажња посвећена је анализи резултата претраге у зависности од различитих конфигурација индекса и упита. Истраживање пружа смернице за имплементацију система за интелигентно претраживање у домену научних публикација.

Кључне речи: *Elasticsearch*, напредна претрага, веб апликација, дигитални документи

Abstract – The paper explores the use of *Elasticsearch* for real-time search optimization on large datasets, with a particular focus on scientific papers from *ArXiv*. It highlights advanced features of *Elasticsearch* such as fuzzy search, synonym processing, and scalability, which improve search efficiency and user experience. The paper also discusses the challenges in optimizing search performance and configurations, offering insights into effective strategies for managing big data. Special attention is paid to the analysis of search results depending on different index and query configurations. The research provides guidelines for the implementation of intelligent search systems in the domain of scientific publications.

Keywords: *Elasticsearch*, advanced search, web application, digital documents

НАПОМЕНА: Овај рад проистекао је из мастер рада чији ментор је био др Милан Челиковић, ванр. проф.

1. УВОД

Услед великог броја података у дигиталном добу, област проналажења информација никад није била важнија. Обим јавно доступних информација је експоненцијално растао заједно са експанзијом употребе интернета и сталним јачањем пословне и

институционалне информационе технологије. Традиционални релациони системи база података суочавају се са ограничењима у обради полу структурираних и неструктурираних докумената, што је омогућило развој *NoSQL* база, попут *Elasticsearch*-а. Ефикасни системи за претрагу комбинују индексирање и претраживање, често користећи инвертовани индекс, метод који је основа многих претраживача, укључујући *Google*. Са порастом корисничких захтева за брзину и тачност, примена напредних алгоритама и дистрибуираних система постала је неопходна.

Elasticsearch је дистрибуирани систем за претрагу и анализу података базиран на библиотеци *Apache Lucene*. Омогућава ефикасну обраду великих количина података, независност од оперативног система и скалабилност. Његова прилагодљивост и *REST API* интерфејс омогућавају laku интеграцију и конфигурацију. Мастер рад се фокусира на оптимизацију претраге у реалном времену за *ArXiv* колекцију научних радова, укључујући генерисање извештаја и обраду агрегационих упита, користећи *Elasticsearch*. Његове могућности укључују подршку за напредне алгоритме, као што су индексирање, рад са синонимима и нејасним упитима, чиме се побољшавају перформансе претраге и задовољство корисника.

2. ПРЕГЛЕД АКТУЕЛНОГ СТАЊА У ОБЛАСТИ

У наставку се даје кратак осврт на најзначајније категорије база података и њихове карактеристике, са нагласком на савремене приступе који омогућавају ефикасну претрагу и анализу података.

2.1. Класификација база података

Данас је на располагању мноштво база података, као и система за управљање истим. Традиционални приступ имплементацији базира се на релационим базама, док модернији приступ представља базу података оријентисану на документе. Оба приступа имају своје случајеве употребе, као и предности и недостатке. Литература пореди претраге текстуалног садржаја у свакој категорији, што подупиरे циљеве рада и проширује знање о савременим техникама претраге.

Релационе базе података чувају податке у табелама које представљају типове ентитета са атрибутима као колонама. Односи између табела дефинишу се преко

примарних и страних кључева, омогућавајући структурисано повезивање.

Нерелационе базе података (NoSQL) користе флексибилне шеме за складиштење података у форматима као што су документи, кључ/вредност или графикони. Оне су оптимизоване за специфичне modele података и модерне апликације, познате по перформансама у раду са великим обимом података.

Базе података базиране на документима складиште полуструктуриране податке у *JSON* формату, омогућавајући флексибилност и једноставно прилагођавање апликацијама. Документи представљају објекте, а колекције докумената се групишу у индексе. Документ-оријентисане базе представљају подскуп *NoSQL* система, али се често издвајају због специфичног модела складиштења и функција претраге.

2.2. Преглед сличне литературе

У оквиру прегледа литературе истраживани су кључни аспекти претраге текстуалних докумената и анализа савремених претраживача и база података. Главни акценат је стављен на поређење перформанси релационих и нерелационих база, као и на анализу претраживача отвореног кода.

„Анализа претраживача отвореног кода“ [1] наглашава важност бесплатних и флексибилних лиценци, стабилности и флексибилности кода, али упозорава на комплексност корисничког интерфејса и могуће додатне трошкове за обуку. Алати *MeiliSearch*, *Typesense*, *Apache Solr* и *Elasticsearch* су поређени у овом раду, а аутори су нагласили да је библиотека иза *Elasticsearch* алата, *Apache Lucene*, најнапреднија на тржишту.

„Поређење перформанси складиштења података за претрагу пуног текста: са структурираним упитним језиком или без?“ [2] анализира перформансе *Elasticsearch*-а и *Solr*-а, истичући њихову брзину и поузданост, уз указивање на предности *PostgreSQL*-а за мање скупове података и на ограничења *MongoDB*.

„Свеобухватна студија *Elasticsearch*-а“ [3] истиче значај инвертованих индекса у *Elasticsearch*-у, као и његову дистрибуирану архитектуру, која обезбеђује поузданост и опоравак од грешака. Аутори су обезбедили детаљан увид у препроцесирање текста које укључује филтере карактера, токенизаторе и филтере токена.

„Управљање дигиталним документима“ [4] је уџбеник који се фокусира на значај релевантности и организације резултата претраге, наглашавајући да корисници траже информације, а не само податке који одговарају упиту.

3. ТРАДИЦИОНАЛНИ ПРИСТУПИ ПРЕТРАЗИ

Системи за претрагу података деле се на централизоване и дистрибуиране, зависно од начина складиштења индекса и обраде упита. Централизовани приступ чува све индексе на једном месту, што поједностављује претрагу, али ограничава скалабилност. Насупрот томе, дистрибуирани индекси омогућавају бржу и скалабилнију претрагу јер су

подаци распоређени по више сервера. Модели претраге укључују класичне и алтернативне моделе. Класични модели претраге су Булов, векторски и пробабилистички, који се разликују по приступу рангирању релевантности. Алтернативни модели су *fuzzy*, проширени Булов, модели неуронских мрежа и језички модели, који уводе напредне функције као што су процена сличности, обрада нејасних упита и контекстуализација.

3.1. Традиционални приступи

Индексирање података користи различите структуре, попут *B* стабала, хеш индекса, бит-мапских и *R* стабала. У *Windows*-у, *NTFS* фајл систем ослања се на *B+* стабла за ефикасно управљање подацима.

Релационе базе попут *PostgreSQL*-а користе напредне технике као што је *GIN* индекс за брзу претрагу текста. Инвертовани индекс, који чува листу докумената за сваки термин, широко се примењује у системима за индексирање текста и интернет претраживачима.

У *NoSQL* базама приступи варирају: од брзих упита у кључ-вредност базама, претраге графова алгоритмима *BFS* и *DFS*, до напредне претраге текста у документ оријентисаним базама, које користе инвертоване индексе за сложене филтере и агрегације.

За претрагу великих текстуалних скупова, дистрибуирани индекси, векторска претрага и документ-оријентисане базе су кључни, нудећи скалабилност, прецизност и ефикасност.

3.2. Алати за претрагу текста

Apache Lucene је *open-source* библиотека за брзо и ефикасно индексирање и претрагу великих количина података, заснована на инвертованим индексима. Она омогућава развој сложених претрага и скалабилних система, али захтева напредне програмерске вештине за оптимизацију.

Apache Solr је платформа за претрагу заснована на *Lucene* библиотеци, дизајнирана за рад са великим количинама структурираних и неструктурираних података. *Solr* додаје функције као што су фасетирана претрага, рангирање и висока доступност, што га чини погодним за примену у великим предузећима и веб апликацијама.

4. МЕХАНИЗМИ ПРЕТРАЖИВАЧА ELASTICSEARCH

Elasticsearch је алат за претрагу и анализу података, базиран на *Apache Lucene* технологији. За разлику од *Lucene* библиотеке, која захтева писање *Java* кода, *Elasticsearch* обезбеђује лакши приступ функцијама претраге и индексирања преко *REST* интерфејса. Подржава концепт скоро реалног времена (*NRT*), што значи да измене у подацима постају претраживе у року од једне секунде.

Иако чува документе, *Elasticsearch* није база података у класичном смислу, већ је оптимизован за *full-text* претрагу и анализу великих количина података. Његова структура се ослања на инвертоване индексе за текстуална поља и *BKD* [5] стабла за нумеричке податке. Поред тога, нуди снажан *Query DSL* језик за креирање напредних упита.

4.1. Терминологија

Кластер је скуп чворова (сервера) који сарађују у складиштењу и обради података, где сваки кластер има јединствено име и један мастер чвор. Чвор је сервер који чува податке и обрађује упите. Партиција је јединица складиштења која омогућава хоризонталну скалабилност, а реплика је копија партиције која побољшава доступност и брзину претраге. Индекси су логичке колекције докумената у *JSON* формату, док су типови у старијим верзијама служили за категоризацију докумената. Документи су основне јединице информација у паровима кључ-вредност, а поља могу бити текстуална или других типова као што су бројеви и датуми, с тим да се текстуална поља обрађују инвертованим индексом.

4.2. Претпроцесирање текста

У *Elasticsearch* претрази се текст мора обрадити пре складиштења, а ова обрада се дешава у фази анализе, односно препроцесирања. Текстуална поља пролазе кроз процес анализе пре складиштења. Компоненте анализатора укључују првенствено филтере карактера, који уклањају непотребне симболе, као што су *HTML* тагови. Затим се користе токенизатори који деле текст на мање јединице - токене, а на крају се примењују токен филтери који нормализују токене, уклањају стоп речи и врше лематизацију. Стандардни анализатор (енг. *Standard analyzer*) је један од многих уграђених анализатора текста у *Elasticsearch*-у.

4.3. Мапирања

Сваки индекс има мапирање или шему за начин на који су поља у документима индексирана. Мапирање се дешава након индексирања и дефинише тип података за свако поље, како поље треба да буде индексирано и како треба да буде ускладиштено. Када се документи додају у *Elasticsearch*, постоје две опције за мапирање. Динамичко мапирање је кад *Elasticsearch* аутоматски детектује типове података и креира мапирања, али не даје увек оптималне резултате и често дупло индексира поједина поља да би се она могла користити за операције филтрирања исто као и за претрагу текста. Експлицитни приступ чине ручно дефинисана мапирања, препоручена за производну употребу. *Elasticsearch* омогућава ефикасну обраду података и напредну претрагу, али захтева пажљиво конфигурисање ради оптималних резултата.

5. ELASTICSEARCH ПРЕТРАГА И СТРАТЕГИЈЕ ОПТИМИЗАЦИЈЕ

Elasticsearch је претраживач текста заснован на технологији *Lucene* који користи токенизацију и стеминг за анализу текста. Подржава специјализоване упите као што су нејасно подударање (*fuzzy query*) и претраге опсега. Упити могу бити комбиновани са филтерима и агрегацијама како би се добила прецизна анализа података.

5.1. Претраживање

Elasticsearch пружа разноврсне функционалности претраге, укључујући претраживање комплетног

текста (путем *match* и *match_phrase* упита), тачно подударање (*term* и *terms* упити за структурирана поља), и упите по опсегу (за бројеве или датуме). Подржава и нејасно подударање (*fuzzy*, *wildcard* и *prefix*), булове комбинације за сложене логичке упите, као и геопросторну претрагу по координатама. Додатно, омогућава анализу са терм-вектор упитима и нуди агрегације и аналитику за статистичку обраду резултата претраге. Ове функције чине *Elasticsearch* ефикасним алатом за претрагу и обраду великих количина података.

5.2. Релевантност документа

Elasticsearch процењује релевантност докумената користећи однос *tf* и *df* мера. Учесталост термина у документу (*tf*) и његова реткост у целокупном корпусу (*idf*) одређују значај термина. У пракси, овај процес оптимизује *BM25* алгоритам [6], који ефикасно мери релевантност за текстуалну претрагу. Документи се затим рангирају према добијеним *score* вредностима, уз примену метрика као што су прецизност и поврат, како би се осигурао најрелевантнији редослед резултата.

5.3. Репликација и партиционисање

Elasticsearch користи партиционисање (енг. *sharding*) за расподелу података на различите сервере, тј. чворове, чиме се побољшава скалабилност и брзина обраде упита. Истовремено, примењује репликацију (енг. *replication*), где се копије података чувају на више чворова, обезбеђујући већу поузданост система и континуирану доступност података чак и у случају отказа појединачних чворова.

5.4. Оптимизације перформанси

Оптимизација перформанси у *Elasticsearch*-у обухвата ефикасно индексирање кроз прецизно дефинисање мапирања података, као што је коришћење типа *keyword* за структуриране податке. Такође, укључује оптимизацију упита употребом филтера и функција попут *search_after* за бржу обраду резултата. Додатно, подешавање кластера и хардвера игра кључну улогу, јер правилна конфигурација сервера и дистрибуција ресурса побољшавају перформансе и скалабилност система.

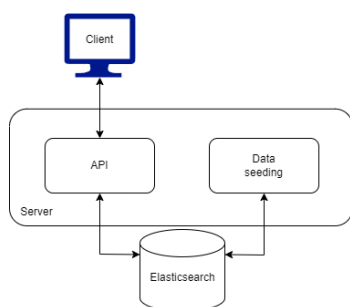
6. АРХИТЕКТУРА СИСТЕМА И МОДЕЛИ ПОДАТАКА

Имплементација веб апликације за претрагу научних радова се састоји од три главна дела: клијентске, серверске апликације и базе података. Подаци се чувају у *JSON* формату, а сви радови су идентификовани преко јединственог *ID*-а. Радови су складиштени у фајл систему.

6.1. Архитектура система

Архитектура, приказана на слици 1, се састоји од четири основна елемента: клијента, серверске апликације, сервера за иницијализацију података и *Elasticsearch* који је коришћен за складиштење и претраживање података. Клијент комуницира са

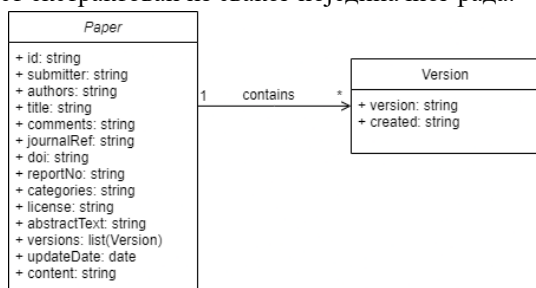
сервером, који обавља различите претраге путем коришћења *QueryDSL Elasticsearch* функционалности.



Слика 1. Архитектура система

6.2. Модел података

ArXiv скуп података садржи метаподатке о научним радовима као што су *ID*, аутори, наслов, апстракт и други који се виде на слици 2. Ови подаци су представљени као *JSON* документи, а садржај радова је доступан као *PDF* путем директног линка. Приликом учитавања података и иницијализације индекса, додатно је поље *content* у којем се складишти индексирани текст екстрактован из сваког појединачног рада.



Слика 2. Дијаграм класа

6.3. Коришћене технологије

У развоју апликације коришћени су различити алати као што су *Spring Boot* за серверску страну, *Python* за препроцесирање података и увоз у *Elasticsearch*, и *React* за креирање фронтенд компонената. *Elasticsearch* је коришћен за складиштење и претрагу података, док *Docker* омогућава изолацију апликација у контејнерима ради боље преносивости и ефикасности.

7. СТУДИЈА СЛУЧАЈА

Апликација за претрагу научних радова има за циљ брзу и релевантну претрагу, оптимизовану за високе перформансе. Основне функције укључују претрагу по текстуалним пољима као што су наслов, аутори, апстракт и садржај радова. Такође, подржава претрагу по подносиоцу научног рада и омогућава генерисање хистограма о броју објављених радова у одређеном периоду. Систем користи *Elasticsearch* за пружање релевантних резултата, а све претраге су интерактивне, освежавајући резултате у реалном времену при промени упита. Резултати претраге укључују наглашене делове текста који одговарају упиту и информације о локацији *PDF* документа. Детаљни приказ докумената обухвата метаподатке, као што су идентификатор рада и име часописа.

Оптимизација система укључује неколико кључних аспеката: *bulk* унос који убрзава индексирање, смањујући време за обраду, екстерно чување *PDF* докумената смањује оптерећење кластеру, а ручно мапирање омогућава боље индексирање метаподатака и текста. *Boosting* побољшава релевантност резултата, док оптимална конфигурација са 2 партиције и 1 репликом даје најбоље перформансе. Прекомерне партиције изазивају пад перформанси, а са већим обимом података, захтеви за хистограмима успоравају претрагу.

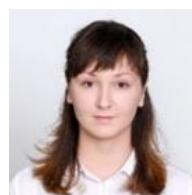
8. ЗАКЉУЧАК

Ова студија истражује претрагу докумената користећи *Elasticsearch*, са фокусом на предности дистрибуираног система за велике скупове података. Обрађене су технике као што су инвертовани индекси, рангирање и партиционисање, уз интеграцију са *Python*-ом, *Spring Boot*-ом и *React.js*-ом за развој веб апликације. *Bulk* операције су убрзале обраду, а оптимална конфигурација са две партиције и једном репликом дала је најбоље резултате, избегавајући проблеме с *oversharding*-ом. Резултати указују на изузетну флексибилност *Elasticsearch*-а и његову примену у разним индустријама. Будућа унапређења укључују анализаторе за српски језик, аутоматско довршавање упита и семантичко рангирање, чиме би се побољшало корисничко искуство и функционалност система.

9. ЛИТЕРАТУРА

- [1] Xinyuan Chang, "The Analysis of Open Source Search Engines", 2023.
- [2] George Fotopoulos, Paris Koloveas, Paraskevi Raftopoulou, Christos Tryfonopoulos, "Comparing Data Store Performance for Full-Text Search: to SQL or to NoSQL?", 2023.
- [3] Nikita Kathare, O. Vinati Reddy, Dr. Vishalakshi Prabhu, "A Comprehensive Study of Elastic Search", 2020.
- [4] Д. Ивановић, Б. Милосављевић, "Управљање дигиталним документима", 2015.
- [5] Octavian Procopiuc, Pankaj K. Agarwal, Lars Arge, Jeffrey Scott Vitter, "Bkd-Tree: A Dynamic Scalable kd-Tree", 2003 .
- [6] Madhusudhan Konda, "Elasticsearch in Action, Second Edition", 2023.

Кратка биографија:



Наталија Шашић рођена је у Сомбору 1999. год. Мастер рад на Факултету техничких наука из области Електротехнике и рачунарства – Софтверско инжењерство и информационе технологије одбранила је 2025 год.
Контакт: sasicnatalija4@gmail.com