Internet je ogromno more prepuno informacija. Problem pronalaska odgovarajućeg skupa informacija je skoro nerešiv bez upotrebe pomoćnih alata, odnosno, mehanizama pretrage. Mehanizmi pretrage predstavljaju moćan alat uz pomoć kojeg se korisnicima nudi skup informacija koje imaju najviše sličnosti sa oktucanim upitom.

***Istorijat***

Univerzitet je pred Alana postavio težak zadatak u to veme, a to je da kreira softver uz pomoć kog je moguće povezati se na Internet. Alan Emtage, administrator sistema za departman informacionih tehnologija i student kanadskog Univerziteta McGill, u septembru 1990. godine objavljuje prvi Internet mehanizam pretrage (*eng. search engine*) pod nazivom *Archie*. Ime *Archie* potiče od reči „*Archive*“, po uzoru na dugačak i mukotrpan proces pretrage kroz koji je Alan prolazio kako bi razvio svoj softver. *Archie* je predstavljao indeks FTP fajlova i njegovi korisnici su uz pomoć jednostavnih upita mogli da ih pretražuju.

Nekoliko godina i mehanizama pretrage kasnije, apsolutnu dominaciju u Internet svetu ima *Yahoo*, čija struktura predstavlja direktorijum koji se sastoji od Internet sajtova, prilikom čega su sajtovi organizovani hijerarhijski. Njegovi kreatori su studenti Stanford Univerziteta, Jerry Yang i David Filo.

1998. godine dvojica studenata, Larry Page i Sergey Brin, koji su takođe sa Stanford Univerziteta odlaze u *Yahoo* sa ponudom da im prodaju svoj biznis za milion dolara, međutim bili su odbijeni. Ono što tada *Yahoo* nije znao jeste da je odbio budućeg giganta koji i dan danas vlada Internetom

Larry i Sergey su želeli da „organizuju sve informacije sveta i da ih učine univerzalno pristupačnim i korisnim“, pa su napravili mehanizam pretrage zasnovan na *PageRank* algoritmu koji istražuje koliko drugih sajtove referencira neku stranicu kako bi se odredila njena relevantnost.

Uzmiajući u obzir prednosti korišćenja mehanizama pretrage i mnogobrojne projekte i pitanja koje studenti postavljaju prilikom njihove izrade, dolazi se do DebugIt-a, platforme koja obuhvata sva implementaciona i diskusiona pitanja studenata, nezavisno od predmeta i tehnologija. Na ovakav način se postiže:

* Svaki student može da pogleda bilo koje pitanje i njegove odgovore, bez obzira na to kada je postavljeno.
* Studenti, asistenti i profesori mogu da odgovore na postavljena pitanja. Na takav način student koji je postavio pitanje brže dolazi do odgovora koji mu je potreban, a asistenti i profesori postaju rasterećeniji od gomile pitanja koje bi inače dobili.
* Najčešći problemi koji se javljaju tokom izrade projekata će vrlo brzo imati odgovore i neće biti potrebe za dupliranjem istih pitanja.
* Studenti se navikavaju da rešenje koje su pronašli prilagode svom kodu.

***Postojeća rešenja***

*Stack Overflow* je sajt, zamišljen da predstavlja biblioteku koja bi umesto knjiga na svojim policama imala sva moguća pitanja vezana za programiranje, sa najkvalitetnijim i vrlo detaljnim odgovorima. Predstavlja samo jedan od 173 sajta koja se nalaze u okviru *Stack Exchange* mreže, koju na mesečnom nivou poseti preko 100 miliona ljudi. Trenutno predstavlja najmoćniju mrežu za pronalaženje odgovora, sa *Stack Overflow* kao vodećim sajtom.

*Discuss The Elastic Stack* predstavlja skup „diskusionih foruma za *Elasticsearch*, *Beats*, *Logstash*, *Kibana*, *Elastic Cloud* i ostale proizvode u okviru *Elastic* ekosistema“.

*GitHub community* je sajt namenjen *GitHub* korisnicima, kao mesto na kome mogu da dobiju odgovor na postavljeno pitanje i da pritom nauče nešto novo ili čak dobiju inspiraciju za neki novi projekat.

Osim navedenih sajtova, koriste se i: Dev.to, Experts Exchange, Code Project i slični.

***Elasticsearch mehanizam pretrage***

*Elasticsearch* je distribuirani, *open source* mehanizam za pretragu i analitiku, čiji temelj predstavlja *Apache Lucene* bibilioteka. Omogućava skladištenje, pretragu i analiziranje velike količine podataka u realnom vremenu.

*Elasticsearch*, danas jedan od najpopularnijih mehanizama pretrage, nastao je kada je Shay Banon, suosnivač *Elastic* kompanije, želeo da kreira mehanizam pretrage koji bi njegova žena mogla da koristi za svoje recepte za kuvanje.

***Struktura Elasticsearch mehanizma pretrage***

Nabrojati samo kao na slici.

***Index***

*Elasticsearch* koristi svoju ugrađenu NoSQL bazu koja podatke čuva kao JSON dokumente. Dokumenti predstavljaju osnovnu jedinicu podatak koju je moguće grupisati u indeks (*eng. index*), koncept sličan tabeli kod relacionih baza podataka. Da bi se dokumenti grupisali u indeks, moraju imati slične karakteristike.

Postoje 2 tipa strukture podataka koje se koriste kod mehanizama pretrage u cilju čuvanja i organizovanja podataka:

* ***forward index*** - dokumenti se mapiraju na termine koje sadrže. To znači da se čuva lista svih reči koje se nalaze u svakom dokumentu. Ova struktura je dobra kada je u pitanju indeksiranje, pošto se svaka reč nadovezuje na prethodnu, ali s druge strane, nije previše efikasna kada je reč o pretrazi na osnovu termina. Na slici se nalazi primer koji prikazuje kako se korišćenjem *forward index* strukture podataka smešta pet dokumenata i njihov sadržaj. Ukoliko korisnik želi da pronađe sve dokumente u kojima se pominje reč „*Elasticsearch*“, mehanizam bi morao da prođe kroz sve dokumente kako bi korisniku vratio rezultat „dokument1, dokument4“.
* ***inverted index*** - termini se mapiraju na dokumente u kojima se nalaze. To znači da za svaku reč postoji lista dokumenata u kojima se nalazi. Kod ovakve strukture je indeksiranje sporije, pošto je za svaku reč potrebno proveriti da li već postoji u indeksu, ali je pretraga veoma brza, čak i kada postoji veoma velika količina podataka, iz razloga što se podaci već unapred čuvaju kao upiti (*eng. queries*). *Elasticsearch* mehanizam pretrage koristi upravo ovu strukturu za smeštanje svojih podataka. Na slici se nalazi primer koji prikazuje kako se korišćenjem *inverted index* strukture podataka smešta pet dokumenata i njihov sadržaj. Ukoliko korisnik želi da pronađe sve dokumente u kojima se pominje reč „Elasticsearch“, mehanizam odmah može da vrati rezultat „dokument1, dokument4“, upravo iz razloga što se podaci čuvaju kao upiti.

***Shard***

*Shard* predstavlja osnovnu gradivnu jedinicu *Elasticsearch* distribuirane arhitekture. Indeks može da sadrži veoma veliku količinu podataka, a bi upravljanje podacima bilo što efikasnije, ti podaci se obično podele u nekoliko *shard* komponenti.

Postoje 2 tipa *shard* komponenti:

* **primarni** (*eng. primary*) - sadrže originalne podatke i obavljaju indeksiranje i pretragu. Broj primarnih *shard* komponenti se definiše prilikom kreiranja indeksa i ne može se naknadno menjati.
* **kopije** (*eng. replica*) - predstavljaju kopiju primarnih *shard* komponenti. Uz pomoć njih se uvodi redudantnost i poboljšavaju se performanse pretrage, jer omogućavaju izvršavanje paralelnih upita. Broj kopija se može dinamički menjati, prilikom čega se automatski balansira broj kopija među node komponentama u cluster komponenti. Preporuka je da svaka primarna *shard* komponenta ima svoju kopiju

***Segment***

Svaka *shard* komponenta se sastoji od više *segment* komponenti, koje imjau *inverted index* strukturu podataka.

Prilikom indeksiranja dokumenta otprilike svake sekunde se kreira novi *segment*.

Sadržaj *segment* komponente se ne može menjati. To znači da svaki put kada se ažurira dokument, taj dokument se zapravo označi kao obrisan i biva obrisan prilikom spajanja *semgnet* komponenti, a umesto njega se indeksira novi sa navedenim izmenama.

Što je više *segment* komponenti, time je pretraga sporija. Zbog toga se vrši spajanje *segment* komponenti istih veličina, počev od manjih. Kada nastane dovoljan broj većih *segment* komponenti iste veličine, onda se vrši njihovo spajanje.

***Cluster i node***

*Node* predstavlja server na kome se izvršava jedna *Elasticsearch* instanca. Jedna ili više *node* komponenti povezanih zajedno čine *cluster*. Svaki *node* u okviru jedne *cluster* komponente zna za ostale *node* komponente, što omogućava preusmeravanje zahteva ka odgovarajućoj *node* komponenti.

*Node* komponente imaju svoje uloge (*eng. roles*) i na osnovu toga se razlikuje više node tipova: master-eligible node, data node, ingest node, remote-eligible node, machine-learning node, transform node, coordinating node.

***Index template i data mapping***

*Index template* je skup podešavanja koja se primenjuju nad *index* komponentom prilikom njenog kreiranja. U okviru njega se definišu brojevi primarnih *shard* komponenti i njihovih kopija, *data mapping*, prioritet i ostalo.

*Data mapping* je šema *index* komponente, odnosno, sadrži informacije o svim poljima i njihovima tipovima koje dokumenti u okviru te *index* komponente imaju. Postoje dva tipa za *data* mapping:

* **statički** - unapred su definisana polja i tipovi,
* **dinamički** - prilikom indeksiranja dokumenata, *Elasticsearch* automatski osvežava (*eng. update*) šemu, kofigurišići imena polja i njihove tipove.

***Alias***

Aliasi se koriste za grupisanje indeksa.