חיפוש OPEN SOURCE במאגרים גדולים (BIGDATA) עם מנוע החיפוש לוסין ו- Elasticsearc

איתמר סין הרשקו

טכנולוגיות חיפוש טקסט התקדמו מאד בתריסר השנים האחרונות. כיום, באמצעות שימוש בכלי חיפוש בקוד פתוח כדוגמת Elasticsearch & Solr, Lucene באפשרותנו לבצע חיפוש יעיל על מאגרי מידע גדולים, גם כאשר כמות המידע אינה יכולה להידחס על שרת אחד. חיפוש על מספר שרתים במקביל, כמו גם כלים מתקדמים לקבלת תובנות ממאגרי המידע אותם אנו מכניסים למנועי החיפוש הופכים את השימוש במנועי חיפוש אלו למשתלם במיוחד.

What is real-world search anyway?

Let’s face it - not all the data we handle is easy to query. In fact, most of it is actually pretty tough to work with. This is often times because a lot of the data we process and handle is unstructured. Be it logs, archived documents, user data, or text fields in our database that we know contain information that can be useful, but we just don’t know how to get to it.

As developers, we tend to fight that. Our first reaction will always be to try and structure the unstructured. This is the challenge we like to rise to as professionals, and that is truly great. But sometimes it just makes sense to stop fighting reality and use a set of tools that is more suited for this task. In some cases this will save many resources and hair-pulling. In other cases, I don’t know whether they are better or worse, we didn't even realize we had a gold mine of information at our fingertips so we haven’t even tried doing something with it.

During the past 10 years or so the field of information retrieval - text retrieval and search engines in particular - has evolved greatly. Search engines have been built and scaled, and within a few years did the impossible. Nobody thought we could handle that scale of data, or to make sense out of it all. Would you have invested in Google before 2000?

Search engines do not exist only on 3rd party websites like Google or Bing. Quite a few search engine libraries that are meant to be used in both open- and closed-source projects were released under various licenses. The most notable of all is probably Apache Lucene (http://lucene.apache.org/), a search engine library released as open-source for the first time in 1999. Since then, Lucene has made giant steps and is developed actively to this day, making new landmarks every few months by releasing new features or major improvements.

But Lucene is just a search library. To scale it out so it can handle large amounts of data you need to have inter-server communications, and some logic to split your data between them. For that Lucene offers Solr, a search server that acts as a wrapper around Lucene indexes. Another option, created by other Lucene project members, is Elasticsearch. Both Solr and Elasticsearch are released under the same open-source license as Lucene's, with my personal favorite being Elasticsearch, due to its novel approach for scaling out indexes and super-easy to use API (everything is doable using REST calls over HTTP).

Using these technologies (Lucene, Solr or Elasticsearch) it is very easy to add full-text search capabilities to any type of application - running on the desktop, web, cloud or mobile. There are a few things to figure out - like how to feed the data from your data sources, how to make sure the search engine has the last version of our data at all times, and how to process it correctly so common searches are effective and perform well. Every project has a different best practice to those challenges, they are hardly ever the same. But once you figured those out, browsing your data is suddenly a breeze.

As it turns out, full-text search capabilities are only the tip of the iceberg. As people started using search technologies to perform full-text searches, new capabilities came about. Leveraging the data and insights search engines can provide on our data, we can do a lot of interesting stuff. For example, we can detect typos and offer corrections; we can find similar documents so we can remove or merge them (also known as record linkage); or we can use this to offer customers at our shop similar products they can add to their cart.

Other, more advanced, modern usages of search technologies that worth noting include geo-spatial search (using shapes like points, circles or polygons representing locations on Earth to find data tagged with more shapes; for example finding the nearest restaurant to the user's location), image search by color scheme (http://blog.qbox.io/boston-elasticsearch-

meetup-scoring-images-by-color), entity extraction and other Natural-Language-Processing methods to further analyze texts and improve insights on them.

There is a great set of tools at our disposal when using search technologies, far more than we can even list in this blog post. Nowadays this is not only about full-text search anymore (although obviously this is definitely still supported and is better than ever before!). Being familiar with those tools and with best practices for using them, we can start giving thought on how we can use them in our project - whether in an automated process or exposed via some UI to our users to give them (and us!) added value.

Modern search engines are built to be scalable and performant. With correct planning you can handle large amounts of data easily (even BigData, if you don't mind the buzzword), as well as many concurrent users issuing many requests, by spreading your data across multiple servers. Because they are so performant, they can offer real-time search capabilities even on large sets of data. The most impressive use of this is most likely Elasticsearch's Kibana (http://demo.kibana.org) dashboard to plot graphs in real-time out of an intensive stream of raw data, for example Apache HTTP server logs.

The field of search engines and information retrieval is moving ahead very fast. There are still many challenges to tackle, but there's already a lot to gain from this quickly evolving set of technologies. Just a quick look at recent history will show you companies that were sold in billions not because they have a great product, but because they were able to collect a lot of data and extract insights out of it.

I'm a search technologies, distributed systems and architecture expert.

Apache Lucene.NET committer, Elasticsearch savant, and the author of RavenDB in Action (http://manning.com/synhershko/).

I'm a frequent speaker at international conferences and provide on-site training and consultancy services around the world.

Currently self-employed as a consultant and freelance developer doing lots of interesting projects world-wide.