



Evolution of Yelp Search to a Ranking Platform

Umesh Dangat

Yelp's Mission

Connecting people with great local businesses.



Yelp by the Numbers

- Our users have written more than 184 million reviews by the end of Q1 2019
- Monthly average of unique visitors who visited Yelp in Q1 2019
 - 35 million via the Yelp app and
 - 69 million via mobile web
- Billions of queries served per year



Yelp Core Search

 Find restaurants Near San Francisco, CA    

 Restaurants  Home Services  Auto Services  More  Write a Review  For Businesses

Best restaurants in San Francisco, CA

 Showing 1-30 of 4387

All Filters

\$

\$\$

\$\$\$

\$\$\$\$

⌚ Open Now

🛵 Order Delivery

📦 Order Takeout

📅 Make a Reservation

📅 Mon, Jan 7

⌚ 7:00 pm

👤 2 people

Find a Table



1. Fog Harbor Fish House (415) 421-2442
★★★★☆ 4896 reviews
\$\$ · Seafood, Bars
Pier 39
📍 Most viewed **Seafood place** in San Francisco
⌚ Current wait time: 0 mins
"Oh my - I've been skipping places like these at the fisherman's wharf for so long, thinking its all..." [read more](#)
[Find a Table](#) Offers reservations



2. Marufuku Ramen SF (415) 872-9786
★★★★☆ 1789 reviews
\$\$ · Ramen
Located in Japan Center
Lower Pacific Heights
📍 Most viewed **Ramen place** in San Francisco
"Omg. What can I say!!!!!! COMPLETELY worth the wait. This review is for the ramen bowls. I love..." [read more](#)



Mo' Map ☐ Redo search when map is moved

Google Data City Map data ©2019 Google Terms of Use

taskrabbit.com ▼

Bay Area Handyman - TaskRabbit - Rates From \$44/Hour. Book Now.

(Ad) Get Skilled **Handymen** To Help Today. Low Rates. Safe & Reliable Service. Over 1 Million Tasks Done. \$1M Happiness Pledge. Background Checked. Cash Free Payment. Services: Minor Repairs, Mounting & Installation, Plumbing, Painting, Yard Work, General **Handyman**.

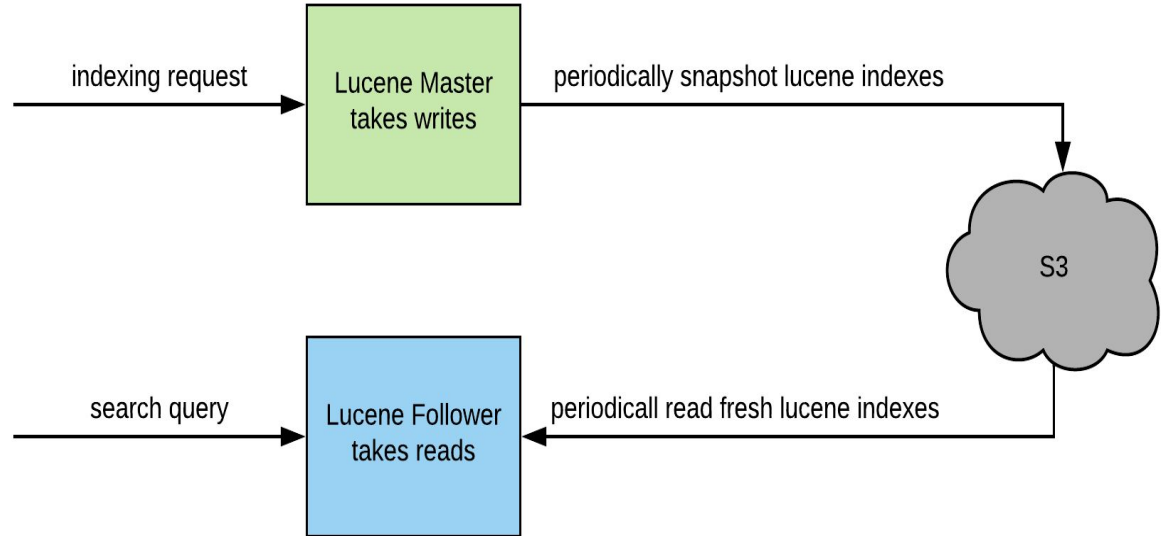
Plumbing Carpentry &



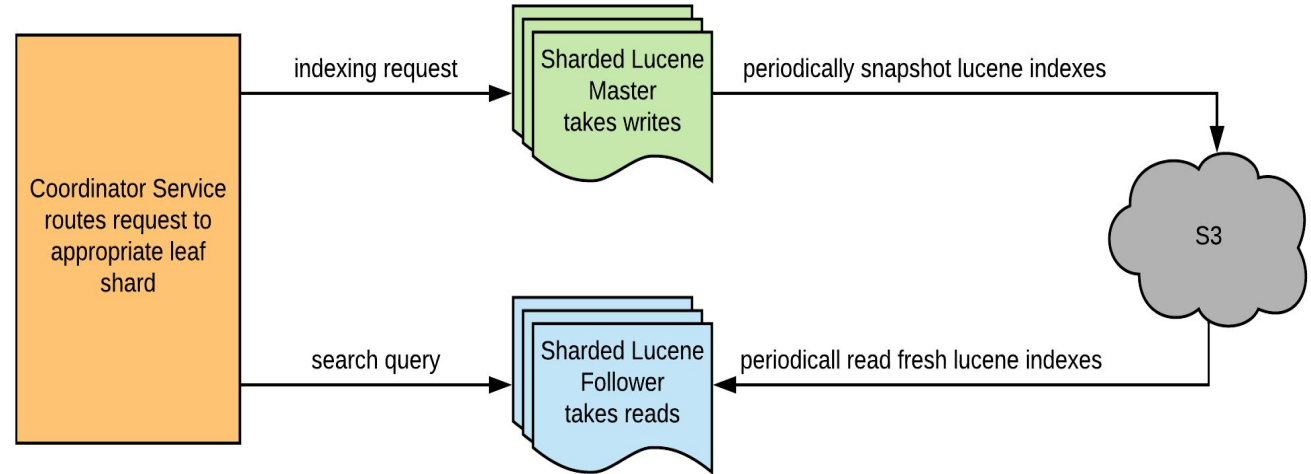
Old Search architecture, until late 2017..



- **Master Follower architecture**
- **leaf search = lucene index**



- distributed lucene indexes (shards)



Issues with this system

Coupled search infrastructure code with search relevance

- Growing team size and specialization of role e.g. relevance experts vs infrastructure engineers implied creating better boundaries in this codebase
- Iteration speed on codebase was getting harder



Issues with this system

Operational burden

- Ever growing size of data meant
 - Many lucene instances due to sharding and replication
 - Slow code pushes
 - Each instance takes time to startup due to data loading
- Engineers spending more time maintaining the system than writing new features



Issues with this system

Adding new features was hard

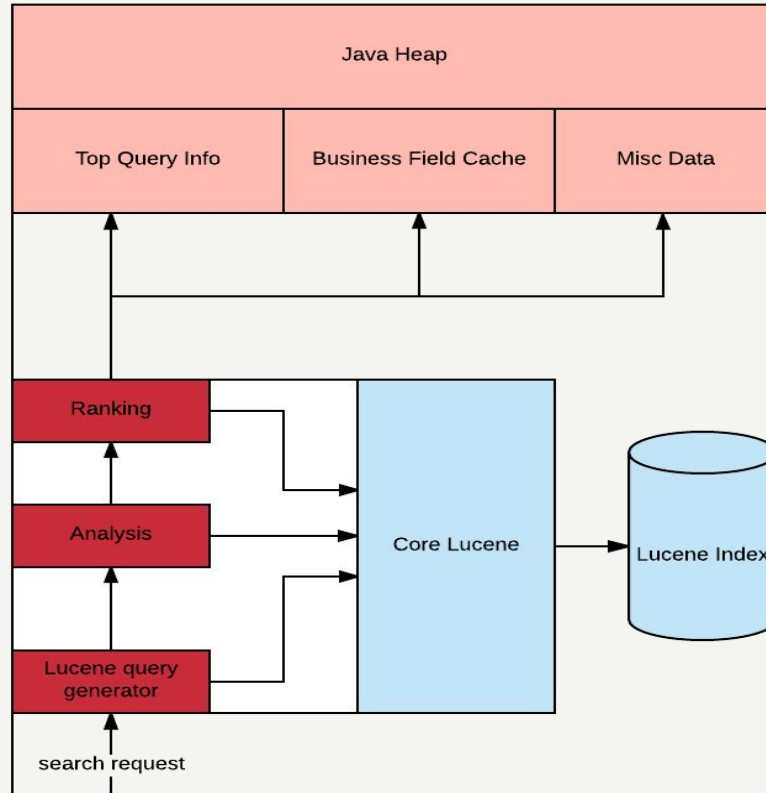
- Modifying schema is hard. So hard to make ranking improvements.
 - Analyzers cannot be iterated upon
 - Adding fields is hard, needs backfill
- Could not do real-time indexing needed for delivery, reservations



Requirements for the new system...



Inside the leaf search node



Components to port

- Custom Ranking (CustomQuery, CustomWeight, CustomScorer)
- Custom Analyzers
- Data on the Java Heap
- Highlights and Logging



**Elasticsearch native plugins allows
us to house custom java code.**



Components to port

- Custom Ranking (CustomQuery, CustomWeight, CustomScorer) → [ScriptPlugin](#)
- Custom Analyzers → [AnalysisPlugin](#)
- Data on the Java Heap → [DocValues](#)
- Other Features:
 - ScoreComponents → [SearchPlugin](#)
 - Highlights → [SearchPlugin](#)



Custom Ranking

Idea is to run the entire scoring code within the `ScriptPlugin`

- Know your scopes. ES provides context at:
 - per document level i.e. `SearchScript.runAsDouble`
 - per segment i.e. `LeafFactory.newInstance(LeafReaderContext)`
 - per shard i.e. `SearchScript.Factory.newFactory` returns `LeafFactory`
 - per query `SearchScript.Factory` instantiation
 - per jvm instance i.e. plugin instantiation
- The class names change with ES versions, but make sure you know the scopes!



Custom Analyzers

Idea is to run the entire analysis code within the [AnalysisPlugin](#)

- Implement *getAnalyzers*
- Issue: Old code uses a much older version of lucene.
 - Quick fix (hack)
 - Use shading
 - Better (time consuming fix)
 - Update version of lucene in your custom analyzers

```
<relocation>  
<pattern>org.apache</pattern>  
<shadedPattern>com.yelp.search.oldschool.analyzers.shaded.org.apache</shadedPattern>  
</relocation>
```



Data off java heap

Majority of the document based data can be moved to the index itself

- SearchScript.getDoc()
- Non document based data can be passed in as query parameters
- Support binary field types in script values
 - [elasticsearch/pull/21484](https://github.com/elastic/elasticsearch/pull/21484)
 - `List<ByteBuffer> queryContext = document.getList(ByteBuffer.class, "query_context");`



Other Features

Implement *FetchSubPhase* to get things about the document like

- Highlights
- Score components/Logging



Performance tips

- Use Elasticsearch tools like [Profile API](#) for measuring bottlenecks
- Within a shard scoring is linear, scales with #shards, *up to a point*
- Doc values FTW !
- Java debugging is “fun”. Using right tools like jprofiler, jmap, jstack helps
 - simply invoking jstack repeatedly helped us find performance bottlenecks
 - CMS was a no go. Have been using G1 in production for over a year now.



Life after generalizing the search architecture



Benefits of the new architecture

- Delete thousands of lines of code needed to manage *federation* of lucene indexes



- Happier Oncall

- Multiple teams at Yelp needing to *filter and rank* could now use the same codebase on different data (ES Clusters) !!

- Ads
- Request a quote

- Unlock new functionality...



ML on ES?

Can we leverage our technological investment in elasticsearch, by hosting machine learning models in elasticsearch?

- Reuse the ES infrastructure
- Scalability and performance
 - Bring computation to each shard
- Reusable infrastructure across teams



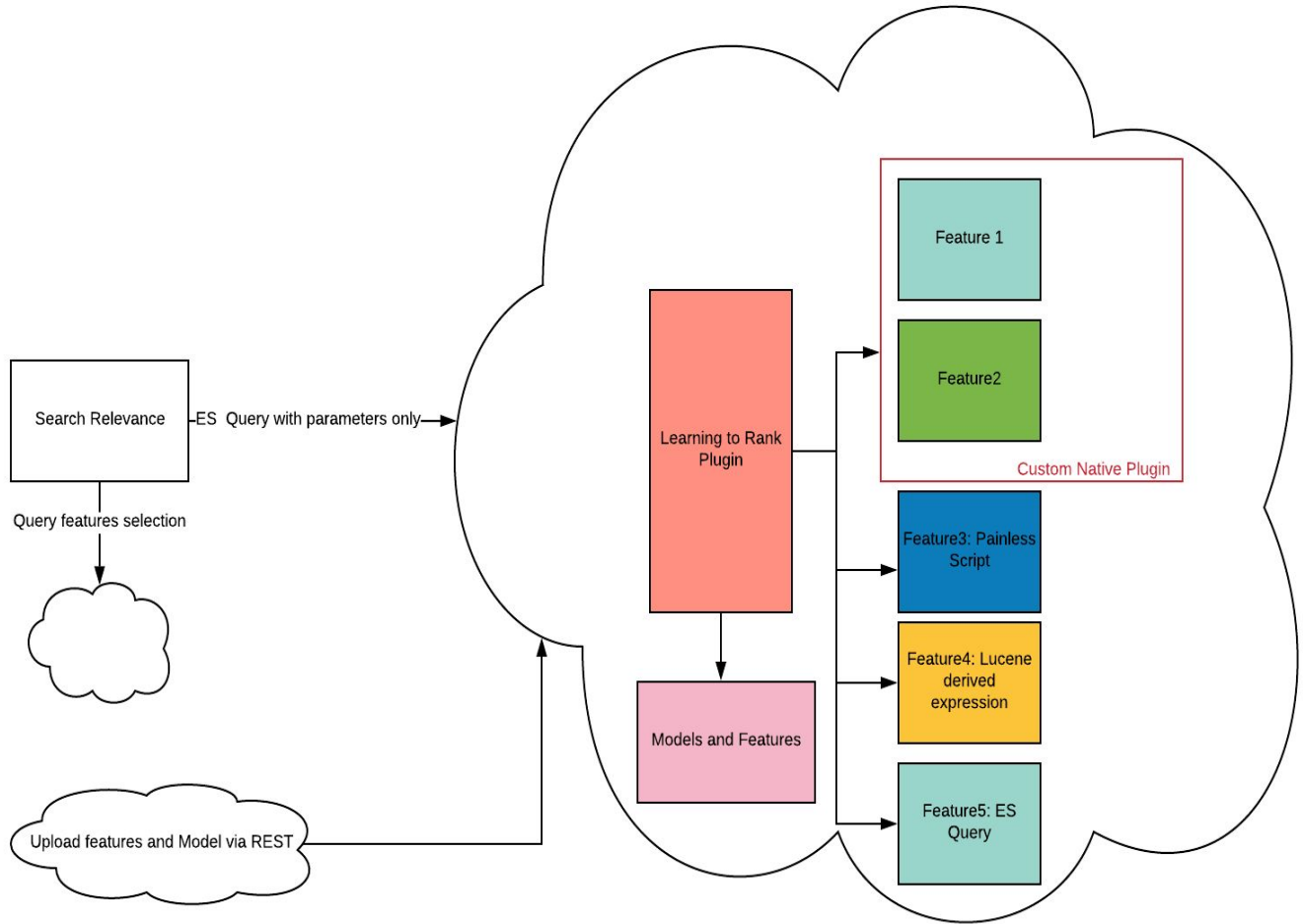
Learning to Rank Plugin

Developed by [Open Source Connections](#)

- Uses elasticsearch to host machine learned models
- Decouple model and feature training from online deployment
 - Simply POST features and models to ES which will be executed query time
- Supports variety of models like linear models, XGBoost, Ranklib



Scoring with LTR



Learning to Rank Plugin and Yelp

- Yelp is a collaborator to the LTR plugin
 - [haystack 2019 talk](#)
- As of today, we run several of our critical search workflows using LTR
- Contributions welcomed!



We're Hiring!

www.yelp.com/careers/

Software Engineer -Data
& Production Backend





fb.com/YelpEngineers



[@YelpEngineering](https://twitter.com/YelpEngineering)



engineeringblog.yelp.com



github.com/yelp



Questions?





Thank you.

