Noise: a search library in Rust

Volker Mische
@vmx

Berlin Buzzwords 2018-06-12, Berlin, Germany

About me

- Volker Mische (@vmx)
- Geo + databases
  - Apache CouchDB (GeoCouch)
  - Couchbase
  - RocksDB
- Rust, Python, JavaScript, Erlang
- “Open Sourcery Coven Leader”

Noise

What is Noise?

- JSON
- Embedded search index/engine
- Full-text, ranges, geo
- Query by example
- https://noisesearch.org/

Demo

Usage

- Insert any JSON
- Automatic indexing:
  - Strings -> Full-text search
  - Numbers -> Ranges
  - GeoJSON -> Bounding boxes
- Query: JSON + operator + value

Technologies

- Node.js API, Python prototype
  (more to come)
- Coded in Rust: Fast, safe, better C/C++
- RocksDB: Key-value store from Facebook

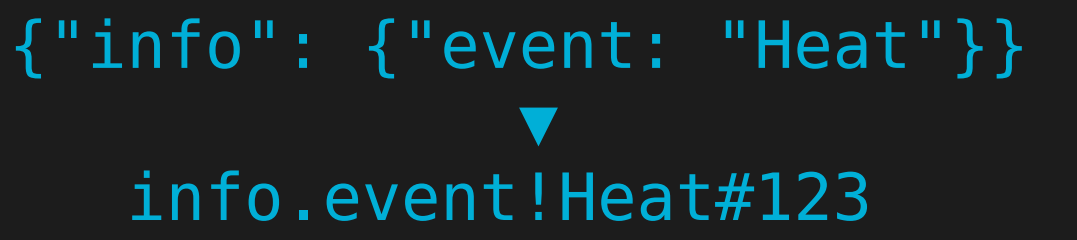
How is it different?

- Traditional FTS: flatten data with mappings
- Document databases with SQL: flatten data when queried
- Noise: preserve structure

How does Noise work?

How does the indexing work?

- Shredder: JSON -> Key-value:



- Indexes are sorted by Internal ID (even the geo index)

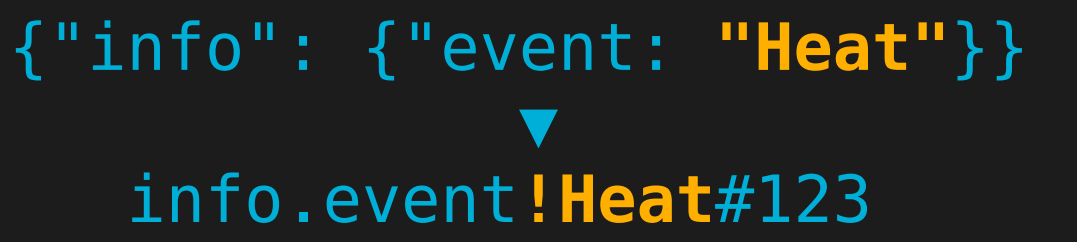
How does the indexing work?

- Shredder: JSON -> Key-value:

- Indexes are sorted by Internal ID (even the geo index)

How does the indexing work?

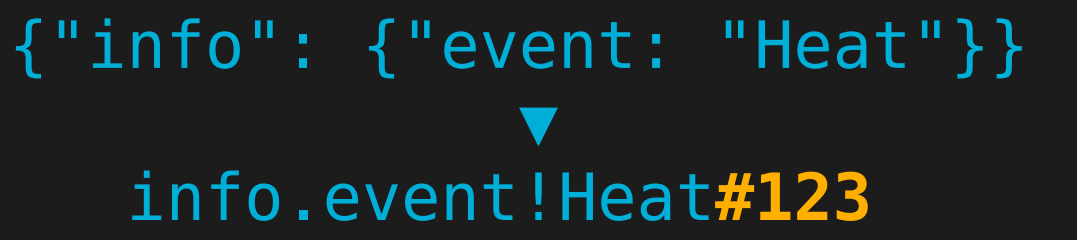
- Shredder: JSON -> Key-value:



- Indexes are sorted by Internal ID (even the geo index)

How does the indexing work?

- Shredder: JSON -> Key-value:



- Indexes are sorted by **Internal ID** (even the geo-index)

How does the indexing work?

…
info.category!Met#1218
…
info.category!Met#1560
…
info.description!high#244
…
info.description!risk#244
…
info.parameter.thunderstorm!möglich#720

How does the indexing work?

…
info.category!Met#1218
…
info.category!Met#1560
…
info.description!high#244
…
info.description!risk#244
…
info.parameter.thunderstorm!möglich#720

How does the indexing work?

…
info.category!Met#1218
…
info.category!Met#1560
…
info.description!high#244
…
info.description!risk#244
…
info.parameter.thunderstorm!möglich#720

Same word, different documents

How does the indexing work?

…
info.category!Met#1218
…
info.category!Met#1560
…
info.description!high#244
…
info.description!risk#244
…
info.parameter.thunderstorm!möglich#720

How does the indexing work?

```
…
info.category!Met#1218
…
info.category!Met#1560
…
info.description!high#244        multiple
…                              words, same
info.description!risk#244        document
…
info.parameter.thunderstorm!möglich#720
```

How does the indexing work?

…
info.category!Met#1218
…
info.category!Met#1560
…
info.description!high#244
…
info.description!risk#244
…
info.parameter.thunderstorm!möglich#720

How does the indexing work?

…
info.category!Met#1218
…
info.category!Met#1560
…
info.description!high#244
…
info.description!risk#244
…
info.parameter.thunderstorm!möglich#720

deeply nested

How does the querying work?

- Similar to traditional DBs (Volcano Iterator Model)
- Parse query
- Build up Query Engine
- Iterate through conditions

Example Query

```
find {
  info: {
    description: ~= "high",
    area: {
      geometry: && [13.1, 52.3, 13.8, 52.7]
    }
  }
}
```

Example Query

```
find {
  info: {
    description: ~= "high",
    area: {
      geometry: && [13.1, 52.3, 13.8, 52.7]
    }
  }
}
```

Example Query

```
find {
  info: {
    description: ~= "high"
  }
}
```

Shred based on operator

Example Query

```
find {
  info: {
    description: ~= "high"
  }
}
```

Shred based on operator

=> info.description!high

Example Query

```
find {
  info: {
    description: ~= "high",
    area: {
      geometry: && [13.1, 52.3, 13.8, 52.7]
    }
  }
}
```

Example Query

```
find {
  info: {
    description: ~= "high",
    area: {
      geometry: && […]
    }
  }
}
```

Example Query

```
info.description!high                    geometry: && […]
```

Example Query

```
info.description!high                    geometry: && […]
```

Example Query

`info.description!high` `geometry: && […]`

…
info.description!few#333
info.description!high#244
info.description!high#333
info.description!high#720
info.description!humidity#244
…

Find first match

Example Query

`info.description!high` geometry: && […]

…
info.description!few#333
▸info.description!high#244
info.description!high#333
info.description!high#720
info.description!humidity#244
…

Found first match

Example Query

`info.description!high` `geometry: && […]`

```
 …
 info.description!few#333
▸info.description!high#244
 info.description!high#333
 info.description!high#720
 info.description!humidity#244
 …
```

Example Query

```
info.description!high                    geometry: && […]
244
```

Example Query

```
info.description!high                    geometry: && […]
244
```

Example Query

info.description!high

244

geometry: && […]

120
184
210
333
387
415

All IIDs within bbox

Example Query

info.description!high
244

geometry: && […]

120
184
210
▸333
387
415

Find IID
>= 244

Example Query

```
info.description!high                    geometry: && […]
244                                      333
```

IIDs don't match => No Result (yet)

Example Query

`info.description!high` geometry: && […]
333

…
info.description!few#333
info.description!high#244
info.description!high#333
info.description!high#720
info.description!humidity#244
…

Find IID >=333

Example Query

info.description!high

geometry: && […]
333

…
info.description!few#333
info.description!high#244
▸info.description!high#333
info.description!high#720
info.description!humidity#244
…

Example Query

```
info.description!high                    geometry: && […]
                                         333

 …
 info.description!few#333
 info.description!high#244
▸info.description!high#333
 info.description!high#720
 info.description!humidity#244
 …
```

Example Query

`info.description!high` geometry: && […]
333

…
info.description!few#333
info.description!high#244
▸info.description!high#**333**
info.description!high#720
info.description!humidity#244
…

Found exact match => Return result

Example Query

`info.description!high` `geometry: && […]`

```
 …
 info.description!few#333
 info.description!high#244
 info.description!high#333
▸info.description!high#720
 info.description!humidity#244
 …
```

Go on with next match

Example Query

info.description!high geometry: && […]

720

…
info.description!few#333
info.description!high#244
info.description!high#333
▸info.description!high#**720**
info.description!humidity#244
…

Example Query

```
info.description!high                    geometry: && […]
720
                                                  120
                                                  184
                                                  210
                                                  333
                                                  387
                                                  415
```

Example Query

info.description!high

720

geometry: && […]

120
184
210
333
387
415

No match
>=720
=> Done

Summary

If one condition narrows down the result a lot, the query should perform well

Future

- Do what people need
- Mapping for including/excluding parts of JSON
- Support for more programming languages
- Scaling up and/or down
- InterPlanetary File System (IPFS)/ InterPlanetary Linked Data (IPLD)

Thanks!

https://try.noisesearch.org/

Volker Mische
http://vmx.cx/
volker.mische@gmail.com
@vmx

Berlin Buzzwords 2018-06-12, Berlin, Germany