Noise: A new search index for semi structured data

Volker Mische
@vmx

Global FOSS4G 2017-08-18, Boston, USA

About me

- Volker Mische (@vmx)
- R-tree implementations for:
  - Apache CouchDB (GeoCouch)
  - Couchbase
  - RocksDB
- Rust, Python, JavaScript, Erlang
- B2G (Boot to Gecko, FirefoxOS)
- Open Source/Open Data proponent

Noise

What is Noise?

- JSON
- Embedded search index/engine
- Query by example
- https://noisesearch.org/

Demo

Usage

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

Technologies

- JavaScript API (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 the indexing work?

- Shredder: JSON -> Key-value:

  `{"info": {"event: "Heat"}}`

  ▼

  `info.event!Heat#123`

- Indexes are sorted by Internal ID (even the R-tree)

How does the indexing work?

- Shredder: JSON -> Key-value:

  `{"info": {"event: "Heat"}}`

  ▼

  `info.event!Heat#123`

- Indexes are sorted by **Internal ID** (even the R-tree)

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: == bbox([10.9, 48.2, 11.0, 48.4])
    }
  }
}
```

Example Query

```
find {
  info: {
    description: ~= "high",
    area: {
      geometry: == bbox([10.9, 48.2, 11.0, 48.4])
    }
  }
}
```

Example Query

```
find {
  info: {
    description: ~= "high",
    area: {
      geometry: == bbox(…)
    }
  }
}
```

Example Query

```
description: ~= "high"      geometry: == bbox(…)
```

Example Query

`description: ~= "high"` `geometry: == bbox(…)`

```
…
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

`description: ~= "high"` `geometry: == bbox(…)`

```
 …
 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

`description: ~= "high"` `geometry: == bbox(…)`

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

Example Query

```
description: ~= "high"          geometry: == bbox(…)
```

244

Example Query

```
description: ~= "high"        geometry: == bbox(…)
244
```

Example Query

`description: ~= "high"`

244

`geometry: == bbox(…)`

120
184
210
333
387
415

All IIDs within bbox

Example Query

description: ~= "high"

244

geometry: == bbox(…)

120
184
210
▸333
387
415

Find IID >= 244

Example Query

```
description: ~= "high"
```
244

```
geometry: == bbox(…)
```
333

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

Example Query

`description: ~= "high"`

`geometry: == bbox(…)`

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

`description: ~= "high"` `geometry: == bbox(…)` **333**

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

Example Query

`description: ~= "high"` `geometry: == bbox(…)`

333

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

Example Query

`description: ~= "high"`

`geometry: == bbox(…)`

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

`description: ~= "high"` `geometry: == bbox(…)`

```
 …
 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

```
description: ~= "high"          geometry: == bbox(…)
720
 …
 info.description!few#333
 info.description!high#244
 info.description!high#333
▸info.description!high#720
 info.description!humidity#244
 …
```

Example Query

```
description: ~= "high"
```

720

```
geometry: == bbox(…)
```

120
184
210
333
387
415

Example Query

`description: ~= "high"`

720

`geometry: == bbox(…)`

120
184
210
333
387
415

No match
=> Done

Summary

If one condition narrows down the result a lot, 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

Thanks!

https://try.noisesearch.org/

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

Global FOSS4G 2017-08-18, Boston, USA