Volltextsuche in Echtzeitdaten mit pg_search

Dr. Marco Hugentobler
Sourcepole AG, Zürich
www.sourcepole.ch

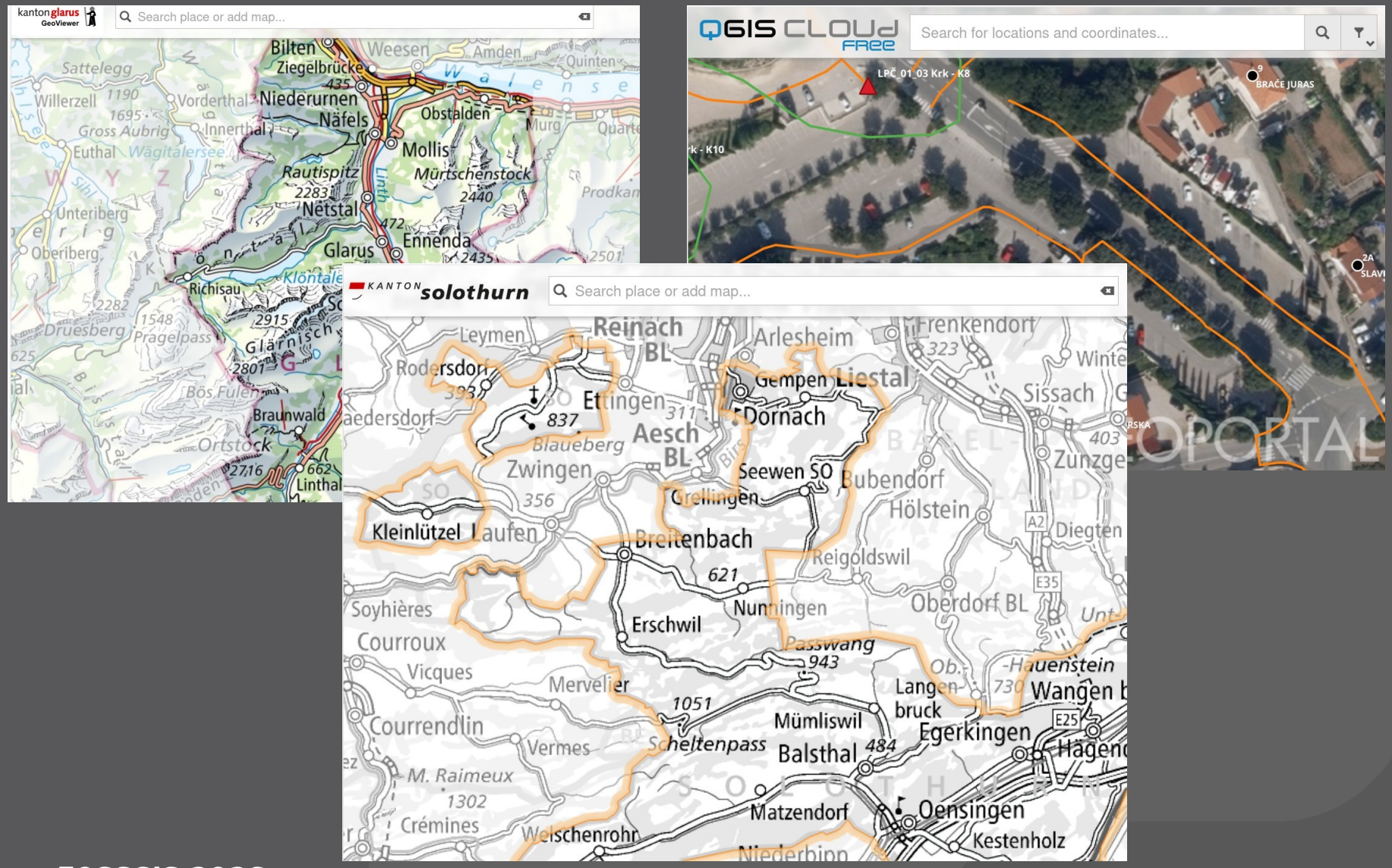
sourcepole
Linux & Open Source Solutions

Sourcepole

- **Web-GIS**
  - QGIS Webclient 2 (QWC2)
  - qgiscloud.com
- **QGIS**
  - 4 C++ Core Entwickler
  - QGIS Server, Drucken, Plugins, ...
  - QGIS LTR (Maintenance und Support)
- **Weitere OSGeo Projekte**
  - Beiträge zu OGR/GDAL, PostGIS, MapServer, Openlayers, OSGeoLive, ...
- **Javascript, Python, C++ & Rust**

Volltextsuche

Fast jede Webkarte hat eine Textsuche



pg_search / ParadeDB

- PostgreSQL extension für Textsuche (bm25)
- Von der Firma ParadeDB entwickelt
- Rust, Lizenz: GNU Affero GPL 3 oder kommerziell
- Unterschied zu PostgreSQL native Suche (tsvector, tsquery):
  - ts_rank beachtet Record isoliert, kein bm25
  - Bessere Performance (gemäss ParadeDB)
- Vorteile gegenüber anderen bm25 Textsuchen:
  - Läuft in PostgreSQL, es braucht keine spezielle Software
  - Index wird live aktualisiert, muss nach Updates der Daten nicht separat generiert werden

Installation

- Docker Image
- Pakete für Debian/Ubuntu und Redhat Linux
- Pakete für MacOS
- Keine Binaries für Windows
- Kompilieren aus dem Quellcode

Grundprinzip Volltextsuche

- Tokenization, Stemming, Normalization
- Inverted index
- Scoring mit bm25 Algorithmus
- Suchoperatoren
- Nur Textsuche, keine semantische Suche

Tokenization

- **Text in Spalte wird in Token aufgeteilt**
- **In pg_search kann der Tokenizer beim Erstellen des Indexes ausgewählt werden:**
  - Unicode (default) : Jedes Wort ist ein Token
  - Ngram (z.B. Trigramm-Suche): Jede Sequenz von drei aufeinanderfolgenden Zeichen ist ein Token
  - Source Code Tokenizer
  - Verschiedene Tokenizer für asiatische Sprachen
- **Jeder Tokenizer hat Optionen (z.B. lowercase, stemming, äöü => aou, löschen von Akzenten)**

Trigram Tokenization

Info | Tabelle | Vorschau | Abfrage (osm_ch_local) ×

Gespeicherte Abfrage | Name | Speichern | Löschen | Dat

```
1 SELECT token from paradedb.tokenize(paradedb.tokenizer('ngram', min_gram => 3, max_gram => 3, prefix_only => false), 'PostgreSQL rocks');
```

Ausführen | 14 Zeilen, 0.003 Sekunden | Sicht erzeugen | Löschen

	token
1	pos
2	ost
3	stg
4	tgr
5	gre
6	res
7	esq
8	sql
9	ql
10	l r
11	ro
12	roc
13	ock
14	cks

Inverted index

- **Keys sind nicht die Texte, sondern die Tokens**

ID	TEXT
1	PostgreSQL supports search
2	Search engines use indexes

Term	Postings List
postgresql	[(1, tf=1, pos=[0])]
supports	[(1, tf=1, pos=[1])]
search	[(1, tf=1, pos=[2]), (2, tf=1, pos=[0])]
engines	[(2, tf=1, pos=[1])]
indexes	[(2, tf=1, pos=[3])]

Bm25 scoring

- **Berücksichtigt Häufigkeit der, aber nicht Nähe oder Reihenfolge**
- **Scoring**
  - Häufigkeit im Text
  - Inverse Häufigkeit im allen Records
  - Länge des Texts

Suchoperatoren und Funktionen

- **||| : Es muss mindestens ein Token enthalten sein**

  description ||| “running shoes”

- **&&& : Es müssen alle Tokens enthalten sein**

  description &&& “running shoes”

- **### : Alle Tokens müssen in der richtigen Reihenfolge enthalten sein**

  description ### “running shoes”

- **=== : Sucht exakte Tokens**

  description === “running”

- **@@@ : Tokens dürfen maximal n Positionen entfernt sein**

  description @@@ (“running” ## 1 ## “shoes”)

OSM-Daten Glarus : Suche in einer Spalte

```sql
DROP INDEX IF EXISTS places_name_idx;
CREATE INDEX places_name_idx ON places USING bm25 (ogc_fid, name) WITH (key_field='ogc_fid');

SELECT name, address_type, city, postcode, housenumber, importance, pdb.score(ogc_fid), address FROM places
WHERE name ||| 'Rüti'
AND (address->>'state' = 'Glarus')
ORDER BY score DESC, importance DESC LIMIT 50;
```

Ausführen | 27 Zeilen, 10.943 Sekunden | Sicht erzeugen | Löschen

	name	address_type	city	postcode	housenumber	importance	score	address
1	Rüti	district	Glarus Süd	8782	*NULL*	0.367367304138263	6.4926147	{ "other": "CH...
2	Rüti	house	Rüti	8782	80	0.19619210675062	6.4926147	{ "other": [...
3	Rüti	district	Glarus Süd	8756	*NULL*	0.13339071107042	6.4926147	{ "other": [...
4	Rüti	locality	Glarus	8750	*NULL*	0.106724044403754	6.4926147	{ "city:de": ...
5	Rüti	locality	Glarus Nord	8874	*NULL*	0.106715018692302	6.4926147	{ "other": [...
6	Rüti	locality	Glarus Nord	8867	*NULL*	0.0800490661478599	6.4926147	{ "other": "CH...
7	Rüti	locality	Glarus Nord	8753	*NULL*	0.0667157328145266	6.4926147	{ "other": "CH...
8	Rüti	locality	Glarus Nord	8753	*NULL*	0.0667157328145266	6.4926147	{ "other": "CH...
9	Rüti	locality	Glarus Nord	8758	*NULL*	0.0667150186923018	6.4926147	{ "other": [...
10	Rüti	locality	Glarus Nord	8865	*NULL*	0.0667150186923018	6.4926147	{ "other": "CH...
11	Rüti	locality	Glarus Nord	8874	*NULL*	0.0667150186923018	6.4926147	{ "other": [...
12	Rüti	locality	Glarus Süd	8782	*NULL*	0.0667147943846799	6.4926147	{ "other": "CH...

Beispiele OSM-Daten Glarus (2)

Trigramm-Suche, zwei Spalten

```sql
DROP INDEX IF EXISTS places_name_idx;

CREATE INDEX places_name_idx ON places
USING bm25 (ogc_fid, (name::pdb.ngram(3,3)), housenumber)
WITH (key_field='ogc_fid');

SELECT name, address_type, city, address->'suburb' AS suburb, postcode, housenumber, importance, pdb.score(ogc_fid), address
FROM places
WHERE (address->>'state' = 'Glarus') AND ( name ||| 'Bahn' OR housenumber ||| '15' )
ORDER BY score DESC, importance DESC
LIMIT 10
```

Ausführen | 10 Zeilen, 17.251 Sekunden | Sicht erzeugen | Löschen

	name	address_type	city	suburb	postcode	housenumber	importance	score	address
1	Bahnhof	house	Glarus Nord	"Beglingen"	8752	29	4.90661478598767e-05	9.717565	{ "other": [
2	Bahnhof	house	Glarus Nord	"Beglingen"	8752	29	4.90661478598767e-05	9.717565	{ "other": [
3	Seilbahn	house	Glarus	"Ennenda"	8755	*NULL*	5.73777370870072e-05	8.856183	{ "other": "(
4	Museum Sernftalbahn	house	Engi	"Engi"	8765	15	4.78149080643476e-05	8.267536	{ "other": [
5	Eisenbahn	house	Glarus Nord	"Beglingen"	8752	49	4.90661478598767e-05	8.1350765	{ "other": [
6	Gumenbahn	house	Glarus Süd	"Braunwald"	8784	*NULL*	4.81277180132298e-05	8.1350765	{ "other": "(
7	Pleusbahn	house	Glarus Süd	"Elm"	8767	*NULL*	4.78149080643476e-05	8.1350765	{ "other": "(
8	Bahnhöfli	house	Nidfurn	"Nidfurn"	8772	6	4.78149080643476e-05	8.1350765	{ "other": [
9	Im Bahnhof	house	Betschwanden	"Marglen"	8777	1	4.81277180132298e-05	7.522558	{ "other": [
10	Sportbahnen	house	Glarus Süd	"Elm"	8767	*NULL*	4.78149080643476e-05	6.995819	{ "other": "(

Beispiele OSM-Daten Glarus (3)

Score gewichten mit pdb.boost

Info | Tabelle | Vorschau | Abfrage (osm_ch_local) ×

Gespeicherte Abfrage | Name

```
1 DROP INDEX IF EXISTS places_name_idx;
2
3 CREATE INDEX places_name_idx ON places
4 USING bm25 (ogc_fid, (name::pdb.ngram(3,3)), housenumber)
5 WITH (key_field='ogc_fid');
6
7 SELECT name, address_type, city, address->'suburb' AS suburb, postcode, housenumber, importance, pdb.score(ogc_fid), address
8 FROM places
9 WHERE (address->>'state' = 'Glarus') AND ( name ||| 'Bahn'::pdb.boost(2) OR housenumber ||| '15' )
10 ORDER BY score DESC, importance DESC
11 LIMIT 10
```

Ausführen | 10 Zeilen, 13.980 Sekunden | Sicht erzeugen | Löschen

	name	address_type	city	suburb	postcode	housenumber	importance	score	address
1	Bahnhof	house	Glarus Nord	"Beglingen"	8752	29	4.90661478598767e-05	19.43513	{ "other": [...
2	Bahnhof	house	Glarus Nord	"Beglingen"	8752	29	4.90661478598767e-05	19.43513	{ "other": [...
3	Seilbahn	house	Glarus	"Ennenda"	8755	*NULL*	5.73777370870072e-05	17.712366	{ "other": "CH...
4	Eisenbahn	house	Glarus Nord	"Beglingen"	8752	49	4.90661478598767e-05	16.270153	{ "other": [...
5	Gumenbahn	house	Glarus Süd	"Braunwald"	8784	*NULL*	4.81277180132298e-05	16.270153	{ "other": "CH...
6	Bahnhöfli	house	Nidfurn	"Nidfurn"	8772	6	4.78149080643476e-05	16.270153	{ "other": [...
7	Pleusbahn	house	Glarus Süd	"Elm"	8767	*NULL*	4.78149080643476e-05	16.270153	{ "other": "CH...
8	Im Bahnhof	house	Betschwanden	"Marglen"	8777	1	4.81277180132298e-05	15.045116	{ "other": [...
9	Sportbahnen	house	Glarus Süd	"Elm"	8767	*NULL*	4.78149080643476e-05	13.991638	{ "other": "CH...
10	Bahnhofplatz	street	Glarus Nord	"Bilten"	8865	*NULL*	0.0533816853589685	13.076036	{ "other": "CH...

OSM-Daten Glarus: Fuzzy-Search

Fuzzy-Suche mit Levenshtein-Distance

```sql
DROP INDEX IF EXISTS places_name_idx;

CREATE INDEX places_name_idx ON places
USING bm25 (ogc_fid, (name::pdb.ngram(3,3)), housenumber)
WITH (key_field='ogc_fid');

SELECT name, address_type, city, address->'suburb' AS suburb, postcode, housenumber, importance, pdb.score(ogc_fid), address
FROM places
WHERE (address->>'state' = 'Glarus') AND name ||| 'Bhnho'::pdb.fuzzy(1)
ORDER BY score DESC, importance DESC
LIMIT 10
```

Ausführen | 10 Zeilen, 14.711 Sekunden | Sicht erzeugen | Löschen

	name	address_type	city	suburb	postcode	housenumber	importance	score	address
1	Bahnhof Mühlehorn	house	Glarus Nord	"Walenguflen"	8874	*NULL*	4.83520256351113e-05	3.0	{ "other": [
2	Mühlehorn Bahnhof	house	Glarus Nord	"Geissegg"	8874	*NULL*	4.83520256351113e-05	3.0	{ "other": [
3	Mühlehorn Bahnhof	house	Glarus Nord	"Walenguflen"	8874	*NULL*	4.83520256351113e-05	3.0	{ "other": [
4	Grünhornhütte	house	Linthal	*NULL*	8783	2	0.167979903475162	2.5	{ "other": [
5	Bahnhofstrasse	street	Glarus	"Ennenda"	8755	*NULL*	0.0533907110704204	2.5	{ "city:de": .
6	Bahnhofstrasse	street	Glarus	"Ennenda"	8755	*NULL*	0.0533907110704204	2.5	{ "city:de": .
7	Bahnhofstrasse	street	Glarus	"Ennenda"	8755	*NULL*	0.0533907110704204	2.5	{ "city:de": .
8	Bahnhofstrasse	street	Glarus	"Ennenda"	8755	*NULL*	0.0533907110704204	2.5	{ "city:de": .
9	Hintere ...	street	Glarus	"Ennenda"	8755	*NULL*	0.0533907110704204	2.5	{ "city:de": .
10	Bahnhofstrasse	street	Glarus	"Ennenda"	8755	*NULL*	0.0533907110704204	2.5	{ "city:de": .

OSM-Daten Glarus: Fuzzy-Search(2)

Mit &&& Operator

Info | Tabelle | Vorschau | Abfrage (osm_ch_local)

Gespeicherte Abfrage | Name

```
1  DROP INDEX IF EXISTS places_name_idx;
2
3  CREATE INDEX places_name_idx ON places
4  USING bm25 (ogc_fid, (name::pdb.ngram(3,3)), housenumber)
5  WITH (key_field='ogc_fid');
6
7  SELECT name, address_type, city, address->'suburb' AS suburb, postcode, housenumber, importance, pdb.score(ogc_fid), address
8  FROM places
9  WHERE (address->>'state' = 'Glarus') AND name &&& 'Bhnho'::pdb.fuzzy(1)
10 ORDER BY score DESC, importance DESC
11 LIMIT 10
```

Ausführen | 10 Zeilen, 13.565 Sekunden | Sicht erzeugen | Löschen

	name	address_type	city	suburb	postcode	housenumber	importance	score	address
1	Bahnhof Mühlehorn	house	Glarus Nord	"Walenguflen"	8874	*NULL*	4.83520256351113e-05	3.0	{ "other": [...
2	Mühlehorn Bahnhof	house	Glarus Nord	"Walenguflen"	8874	*NULL*	4.83520256351113e-05	3.0	{ "other": [...
3	Mühlehorn Bahnhof	house	Glarus Nord	"Geissegg"	8874	*NULL*	4.83520256351113e-05	3.0	{ "other": [...
4	Bahnhofstrasse	street	Glarus	"Ennenda"	8755	*NULL*	0.0533907110704204	2.5	{ "city:de": ...
5	Bahnhofstrasse	street	Glarus	"Ennenda"	8755	*NULL*	0.0533907110704204	2.5	{ "city:de": ...
6	Bahnhofstrasse	street	Glarus	"Ennenda"	8755	*NULL*	0.0533907110704204	2.5	{ "city:de": ...
7	Bahnhofstrasse	street	Glarus	"Ennenda"	8755	*NULL*	0.0533907110704204	2.5	{ "city:de": ...
8	Bahnhofstrasse	street	Glarus	"Ennenda"	8755	*NULL*	0.0533907110704204	2.5	{ "city:de": ...
9	Hintere Bahnhofstrasse	street	Glarus	"Ennenda"	8755	*NULL*	0.0533907110704204	2.5	{ "city:de": ...
10	Bahnhofstrasse	street	Glarus	"Ennenda"	8755	*NULL*	0.0533907110704204	2.5	{ "city:de": ...

Referenzen

- https://docs.paradedb.com/documentation
- https://www.paradedb.com/learn/search-concepts/full-text-search
- https://www.paradedb.com/blog/agpl

Vielen Dank für die Aufmerksamkeit

Marco Hugentobler
marco.hugentobler @ s o u r c e p o l e . c h