PolyGlot-Database Performance

Benchmark Framework - MongoDB vs Neo4J

Hyeon Ung Kim, Tim Niehoff

6. August 2018

Aufgabenstellung

Heranführung

- Verschiedene DB Typen mit unterschiedlichen Vorteilen[1]

[1] Inhalt der Folie von
https://dbs.uni-leipzig.de/file/Intro_bdprak_final.pdf

Heranführung

- Verschiedene DB Typen mit unterschiedlichen Vorteilen[1]
  - Relational: Sicherheit, homogene Daten

[1] Inhalt der Folie von
https://dbs.uni-leipzig.de/file/Intro_bdprak_final.pdf

Heranführung

- Verschiedene DB Typen mit unterschiedlichen Vorteilen[1]
  - Relational: Sicherheit, homogene Daten
  - Document: Flexibles Schema, Suchfunktionen

[1] Inhalt der Folie von
https://dbs.uni-leipzig.de/file/Intro_bdprak_final.pdf

Heranführung

- Verschiedene DB Typen mit unterschiedlichen Vorteilen[1]
  - Relational: Sicherheit, homogene Daten
  - Document: Flexibles Schema, Suchfunktionen
  - Graph: Beziehungen, Traversal

[1] Inhalt der Folie von
`https://dbs.uni-leipzig.de/file/Intro_bdprak_final.pdf`

Heranführung

- Verschiedene DB Typen mit unterschiedlichen Vorteilen[1]
  - Relational: Sicherheit, homogene Daten
  - Document: Flexibles Schema, Suchfunktionen
  - Graph: Beziehungen, Traversal
- Polyglot: Verwendung mehrerer DB-Typen für untersch. Anwendungsfälle

[1] Inhalt der Folie von
`https://dbs.uni-leipzig.de/file/Intro_bdprak_final.pdf`

Heranführung

- Verschiedene DB Typen mit unterschiedlichen Vorteilen[1]
  - Relational: Sicherheit, homogene Daten
  - Document: Flexibles Schema, Suchfunktionen
  - Graph: Beziehungen, Traversal
- Polyglot: Verwendung mehrerer DB-Typen für untersch. Anwendungsfälle
- Aufgabe: Vergleich einer Graphdatenbank mit einer Dokumenten-Datenbank

[1] Inhalt der Folie von
`https://dbs.uni-leipzig.de/file/Intro_bdprak_final.pdf`

gegebene Werkzeuge

Resultat: PolyGDBP

Resultat: PolyGDBP

Features

- Command line Interface mit Parametereingabe (Query, Serveradressen, Datensatz, ...)

- Command line Interface mit Parametereingabe (Query, Serveradressen, Datensatz, ...)
- Obligatorisch nur Queryangabe

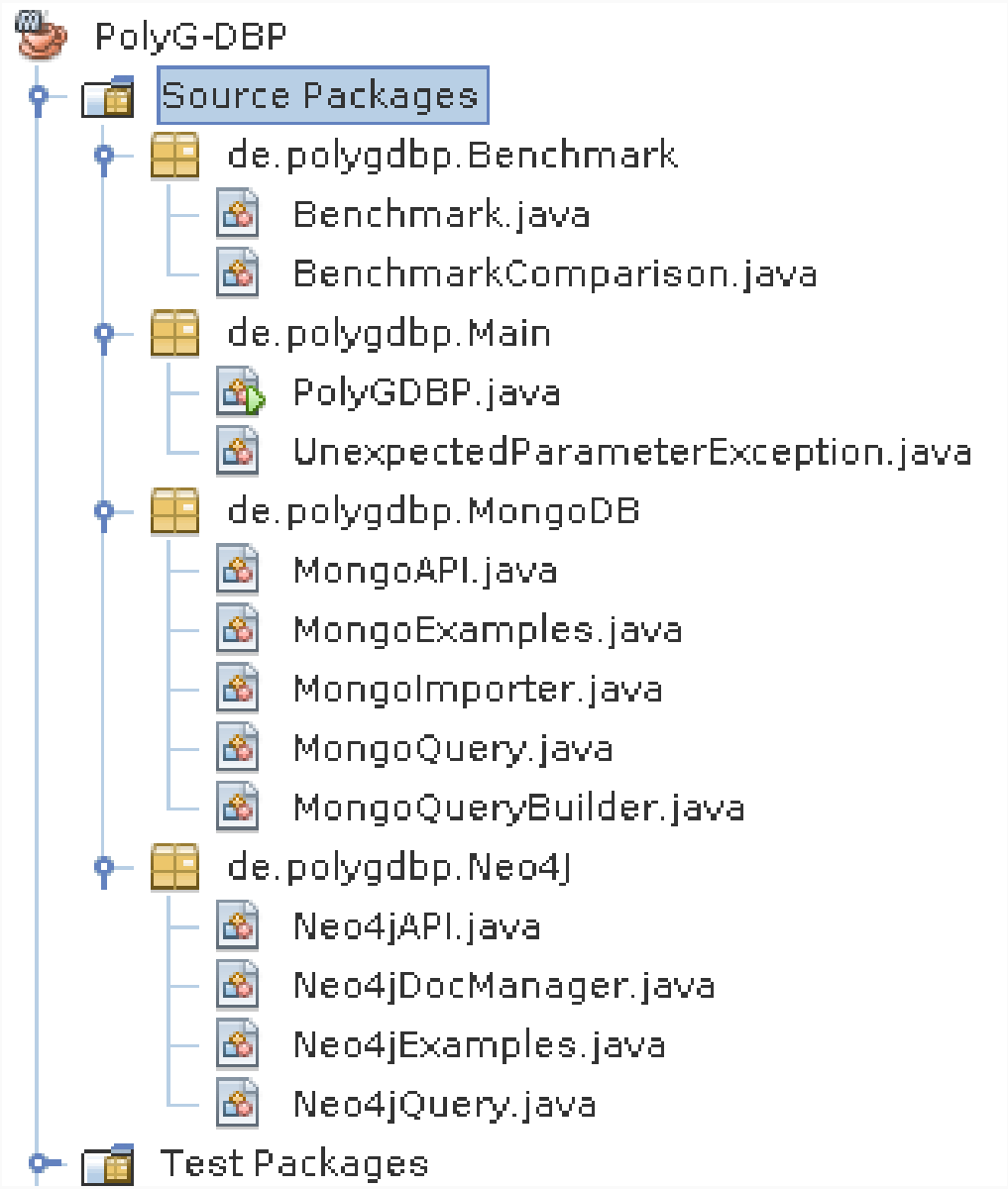
- Command line Interface mit Parametereingabe (Query, Serveradressen, Datensatz, ...)
- Obligatorisch nur Queryangabe
- Vorgefertigte Queries für Yelp Datensatz, ansonsten Angabe benutzerspezifischer Queries

- Command line Interface mit Parametereingabe (Query, Serveradressen, Datensatz, ...)
- Obligatorisch nur Queryangabe
- Vorgefertigte Queries für Yelp Datensatz, ansonsten Angabe benutzerspezifischer Queries
- Reduzieren und Einlesen des Datensatzes möglich (via MongoAPI und MongoConnector)

- Command line Interface mit Parametereingabe (Query, Serveradressen, Datensatz, ...)
- Obligatorisch nur Queryangabe
- Vorgefertigte Queries für Yelp Datensatz, ansonsten Angabe benutzerspezifischer Queries
- Reduzieren und Einlesen des Datensatzes möglich (via MongoAPI und MongoConnector)
- Vor und nach jedem Schritt Stoppen der Zeit, Logging, Filewriting

- Command line Interface mit Parametereingabe (Query, Serveradressen, Datensatz, ...)
- Obligatorisch nur Queryangabe
- Vorgefertigte Queries für Yelp Datensatz, ansonsten Angabe benutzerspezifischer Queries
- Reduzieren und Einlesen des Datensatzes möglich (via MongoAPI und MongoConnector)
- Vor und nach jedem Schritt Stoppen der Zeit, Logging, Filewriting
- Visualisierung mittels Javascript Anwendung möglich

Programmstruktur



Tests mit dem Yelp Dataset

Tests mit dem Yelp Dataset

Datensatz + DatenModell

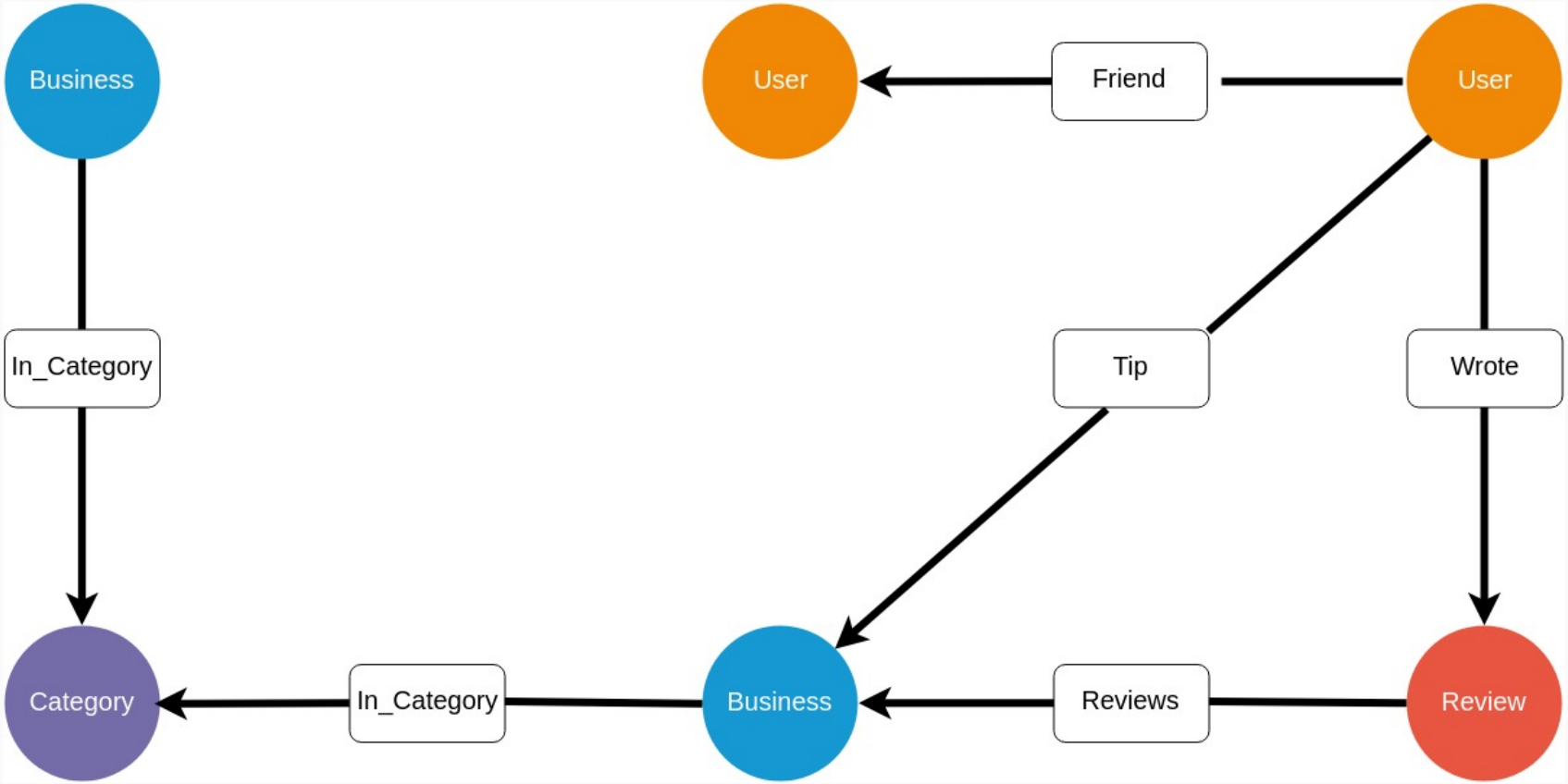
- Yelp = Suchmaschine und Empfehlungsportal für Restaurants und Geschäfte

- Yelp = Suchmaschine und Empfehlungsportal für Restaurants und Geschäfte
- Datensatz $\geq$ 6,5 GB

- Yelp = Suchmaschine und Empfehlungsportal für Restaurants und Geschäfte
- Datensatz $\geq$ 6,5 GB
- besteht aus 6 *.jsons: business, checkin, photos, review, tip, user

- Yelp = Suchmaschine und Empfehlungsportal für Restaurants und Geschäfte
- Datensatz $\geq$ 6,5 GB
- besteht aus 6 *.jsons: business, checkin, photos, review, tip, user
-
```
{"business_id": "FYWN1wneV18bWNgQjJ2GNg", "name": "Dental by Design", "neighborhood":
"", "address": "4855 E Warner Rd, Ste B9", "city": "Ahwatukee", "state": "AZ",
"postal_code": "85044", "latitude": 33.3306902, "longitude": -111.9785992, "stars": 4.0,
"review_count": 22, "is_open": 1, "attributes": {"AcceptsInsurance": true,
"ByAppointmentOnly": true, "BusinessAcceptsCreditCards": true}, "categories":
["Dentists", "General Dentistry", "Health & Medical", "Oral Surgeons", "Cosmetic
Dentists", "Orthodontists"], "hours": {"Friday": "7:30-17:00", "Tuesday": "7:30-17:00",
"Thursday": "7:30-17:00", "Wednesday": "7:30-17:00", "Monday": "7:30-17:00"}}
```

Neo4j Datenmodell



Tests mit dem Yelp Dataset

Ergebnisse

Zusammenfassung

Zusammenfassung

- PolyG-DBP = Framework zum Vergleich einer Dokument-DB mit einer Graphdatenbank

Zusammenfassung

- PolyG-DBP = Framework zum Vergleich einer Dokument-DB mit einer Graphdatenbank
- konkret: MongoDB und Neo4j werden getestet

Zusammenfassung

- PolyG-DBP = Framework zum Vergleich einer Dokument-DB mit einer Graphdatenbank
- konkret: MongoDB und Neo4j werden getestet
- Ausführung von prebuilt Queries oder nutzerspezifischen Queries

Zusammenfassung

- PolyG-DBP = Framework zum Vergleich einer Dokument-DB mit einer Graphdatenbank
- konkret: MongoDB und Neo4j werden getestet
- Ausführung von prebuilt Queries oder nutzerspezifischen Queries
- Messen und Vergleich der Query-Ausführungszeiten

Zusammenfassung

- PolyG-DBP = Framework zum Vergleich einer Dokument-DB mit einer Graphdatenbank
- konkret: MongoDB und Neo4j werden getestet
- Ausführung von prebuilt Queries oder nutzerspezifischen Queries
- Messen und Vergleich der Query-Ausführungszeiten
- Vergleich Neo4j und MongoDB mit Yelp Datensatz: MongoDB performanter bei einfachen Queries, Neo4j punktet bei Queries mit mehreren Relationen/Collections

Ausblick

Zukünftige Fragestellungen

- Optimierung der Datenbanken (z.B. Indexierung)

Zukünftige Fragestellungen

- Optimierung der Datenbanken (z.B. Indexierung)
- Auswirkungen des konstruierten Neo4j Datenmodells auf Queryzeit

Zukünftige Fragestellungen

- Optimierung der Datenbanken (z.B. Indexierung)
- Auswirkungen des konstruierten Neo4j Datenmodells auf Queryzeit
- Betrachte neben Queryzeit die Synchronisationszeit zwischen MongoDB und Neo4j

Fragen?