

PolyGlot-Database Performance

Framework zum Vergleichen von Anfragedauern auf
MongoDB und Neo4J

Hyeon Ung Kim, Tim Niehoff

6. August 2018

Aufgabenstellung

- Verschiedene Anwendungen erfordern versch. Typen von Datenbanken: Relational, Key-Value, Document, Graph¹

¹Inhalt der Folie von

https://dbs.uni-leipzig.de/file/Intro_bdprak_final.pdf

Heranführung

- Verschiedene Anwendungen erfordern versch. Typen von Datenbanken: Relational, Key-Value, Document, Graph¹
- In der Praxis: Gleichzeitige Verwendung versch. Typen

¹Inhalt der Folie von

https://dbs.uni-leipzig.de/file/Intro_bdprak_final.pdf

Heranführung

- Verschiedene Anwendungen erfordern versch. Typen von Datenbanken: Relational, Key-Value, Document, Graph¹
- In der Praxis: Gleichzeitige Verwendung versch. Typen
- Vorteil: Optimale DB für jeden Anwendungsfall

¹Inhalt der Folie von

https://dbs.uni-leipzig.de/file/Intro_bdprak_final.pdf

Heranführung

- Verschiedene Anwendungen erfordern versch. Typen von Datenbanken: Relational, Key-Value, Document, Graph¹
- In der Praxis: Gleichzeitige Verwendung versch. Typen
- Vorteil: Optimale DB für jeden Anwendungsfall
 - Relational: Sicherheit, homogene Daten

¹Inhalt der Folie von

https://dbs.uni-leipzig.de/file/Intro_bdprak_final.pdf

Heranführung

- Verschiedene Anwendungen erfordern versch. Typen von Datenbanken: Relational, Key-Value, Document, Graph¹
- In der Praxis: Gleichzeitige Verwendung versch. Typen
- Vorteil: Optimale DB für jeden Anwendungsfall
 - Relational: Sicherheit, homogene Daten
 - Document: Flexibles Schema, Suchfunktionen

¹Inhalt der Folie von

https://dbs.uni-leipzig.de/file/Intro_bdprak_final.pdf

Heranführung

- Verschiedene Anwendungen erfordern versch. Typen von Datenbanken: Relational, Key-Value, Document, Graph¹
- In der Praxis: Gleichzeitige Verwendung versch. Typen
- Vorteil: Optimale DB für jeden Anwendungsfall
 - Relational: Sicherheit, homogene Daten
 - Document: Flexibles Schema, Suchfunktionen
 - Graph: Beziehungen, Traversal

¹Inhalt der Folie von

https://dbs.uni-leipzig.de/file/Intro_bdprak_final.pdf

Heranführung

- Verschiedene Anwendungen erfordern versch. Typen von Datenbanken: Relational, Key-Value, Document, Graph¹
- In der Praxis: Gleichzeitige Verwendung versch. Typen
- Vorteil: Optimale DB für jeden Anwendungsfall
 - Relational: Sicherheit, homogene Daten
 - Document: Flexibles Schema, Suchfunktionen
 - Graph: Beziehungen, Traversal
- Aufgabe: Welchen Vorteil hat die Verwendung einer Graphdatenbank gegenüber einer Dokumenten-Datenbank?

¹Inhalt der Folie von

https://dbs.uni-leipzig.de/file/Intro_bdprak_final.pdf

gegebene Werkzeuge



Stichpunkte zur vorigen Folie

- Mongo + Neo4j Driver

Stichpunkte zur vorigen Folie

- Mongo + Neo4j Driver
- teilweise nicht ausgereift. Für das garantieren von beliebigen Queries muss geparst werden

**Problem: Gleicher Datensatz
auf beiden DB?**

**Problem: Gleicher Datensatz
auf beiden DB?**

**Mongo Connector + Neo4j Doc
Manager**

Mongo Connector + Neo4j Doc Manager

- Softwarelösung von MongoDB und Neo4j
- synchronisiert mithilfe von Mongo Replica Sets von Mongo in Neo4j
- muss kein Datenmodell angegeben werden. Funzt Generisch

**Problem: Gleicher Datensatz
auf beiden DB?**

Apoc

- Software von Neo4j zum Import von JSONs in die Neo4j

- Software von Neo4j zum Import von JSONs in die Neo4j
- keine Synchronisierung zwischen Mongo und Neo4j

- Software von Neo4j zum Import von JSONs in die Neo4j
- keine Synchronisierung zwischen Mongo und Neo4j
- Nutzer muss Neo4j Datenmodell erstellen

- Software von Neo4j zum Import von JSONs in die Neo4j
- keine Synchronisierung zwischen Mongo und Neo4j
- Nutzer muss Neo4j Datenmodell erstellen
- durch nutzerspezifisches Datenmodell kann man die Graphdatenbankvorteile auskosten

Resultat: PolyGDBP

Resultat: PolyGDBP

Pipline und Workflow

- File Struktur (Main, Mongo, Neo4j, Benchmark)

- File Struktur (Main, Mongo, Neo4j, Benchmark)
- Command line Interface

- File Struktur (Main, Mongo, Neo4j, Benchmark)
- Command line Interface
- nur Queryangabe ist pflicht, alles andere optional.

- File Struktur (Main, Mongo, Neo4j, Benchmark)
- Command line Interface
- nur Queryangabe ist pflicht, alles andere optional.
- gibt vorgefertigte Queries für Yelp Datensatz

- File Struktur (Main, Mongo, Neo4j, Benchmark)
- Command line Interface
- nur Queryangabe ist pflicht, alles andere optional.
- gibt vorgefertigte Queries für Yelp Datensatz
- Nutzer kann JSON Datensatz importieren

- File Struktur (Main, Mongo, Neo4j, Benchmark)
- Command line Interface
- nur Queryangabe ist pflicht, alles andere optional.
- gibt vorgefertigte Queries für Yelp Datensatz
- Nutzer kann JSON Datensatz importieren
- Wird in MongoCollections verarbeitet und durch MongoConnector erfolgt danach das Übertragen in Neo4j

- File Struktur (Main, Mongo, Neo4j, Benchmark)
- Command line Interface
- nur Queryangabe ist pflicht, alles andere optional.
- gibt vorgefertigte Queries für Yelp Datensatz
- Nutzer kann JSON Datensatz importieren
- Wird in MongoCollections verarbeitet und durch MongoConnector erfolgt danach das Übertragen in Neo4j
- Queries werden ausgeführt

- File Struktur (Main, Mongo, Neo4j, Benchmark)
- Command line Interface
- nur Queryangabe ist pflicht, alles andere optional.
- gibt vorgefertigte Queries für Yelp Datensatz
- Nutzer kann JSON Datensatz importieren
- Wird in MongoCollections verarbeitet und durch MongoConnector erfolgt danach das Übertragen in Neo4j
- Queries werden ausgeführt
- vor und jedem dieser Schritte zeit gestoppt, geloggt und schließlich ausgegeben

- File Struktur (Main, Mongo, Neo4j, Benchmark)
- Command line Interface
- nur Queryangabe ist pflicht, alles andere optional.
- gibt vorgefertigte Queries für Yelp Datensatz
- Nutzer kann JSON Datensatz importieren
- Wird in MongoCollections verarbeitet und durch MongoConnector erfolgt danach das Übertragen in Neo4j
- Queries werden ausgeführt
- vor und jedem dieser Schritte zeit gestoppt, geloggt und schließlich ausgegeben
- kann mithilfe unserer Javascript Anwendung visualisiert werden.

**Lets run it. With the help of
YELP**

**Lets run it. With the help of
YELP**

Datensatz + DatenModell

- YELP = Suchmaschine und Empfehlungsportal für Restaurants und Geschäfte

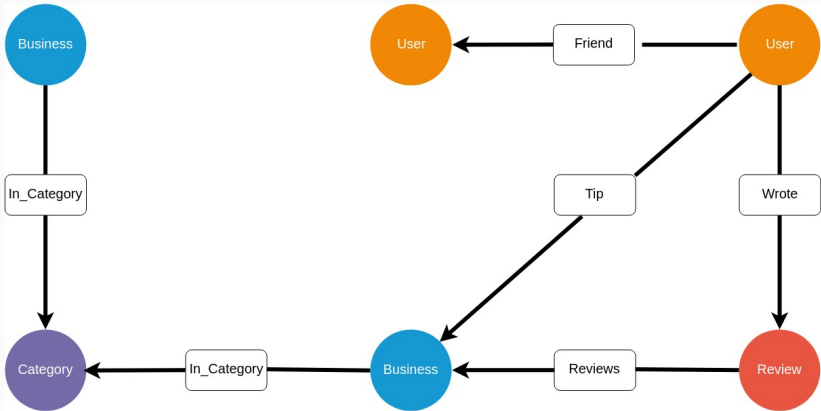
- YELP = Suchmaschine und Empfehlungsportal für Restaurants und Geschäfte
- YELP Datensatz X GB groß

- YELP = Suchmaschine und Empfehlungsportal für Restaurants und Geschäfte
- YELP Datensatz X GB groß
- besteht aus X .jsons: users, business, review, ...

- YELP = Suchmaschine und Empfehlungsportal für Restaurants und Geschäfte
- YELP Datensatz X GB groß
- besteht aus X .jsons: users, business, review, ...
- Beispiele eines JSON Business Objektes:

- YELP = Suchmaschine und Empfehlungsportal für Restaurants und Geschäfte
- YELP Datensatz X GB groß
- besteht aus X .jsons: users, business, review, ...
- Beispiele eines JSON Business Objektes:
 - ...

Neo4j Datenmodell



**Lets run it. With the help of
YELP**

Ergebnisse

Zusammenfassung

- Verschiedene DB-Typen haben versch. Vor- und Nachteile

Zusammenfassung

- Verschiedene DB-Typen haben versch. Vor- und Nachteile
- PolyG-DBP ist ein Framework zum Vergleich einer Dokument-DB mit einer Graphdatenbank

Zusammenfassung

- Verschiedene DB-Typen haben versch. Vor- und Nachteile
- PolyG-DBP ist ein Framework zum Vergleich einer Dokument-DB mit einer Graphdatenbank
- konkret: MongoDB und Neo4j werden getestet

Zusammenfassung

- Verschiedene DB-Typen haben versch. Vor- und Nachteile
- PolyG-DBP ist ein Framework zum Vergleich einer Dokument-DB mit einer Graphdatenbank
- konkret: MongoDB und Neo4j werden getestet
- Nutzer kann prebuilt queries ausführen lassen oder eigene Queries testen lassen

Zusammenfassung

- Verschiedene DB-Typen haben versch. Vor- und Nachteile
- PolyG-DBP ist ein Framework zum Vergleich einer Dokument-DB mit einer Graphdatenbank
- konkret: MongoDB und Neo4j werden getestet
- Nutzer kann prebuilt queries ausführen lassen oder eigene Queries testen lassen
- Ausführungszeiten der Queries werden gemessen und verglichen

Zusammenfassung

- Verschiedene DB-Typen haben versch. Vor- und Nachteile
- PolyG-DBP ist ein Framework zum Vergleich einer Dokument-DB mit einer Graphdatenbank
- konkret: MongoDB und Neo4j werden getestet
- Nutzer kann prebuilt queries ausführen lassen oder eigene Queries testen lassen
- Ausführungszeiten der Queries werden gemessen und verglichen
- Unsere Tests mithilfe von PolyG-DBP und Yelp zeigen: Neo4j und MongoDB siegen bei bestimmten Arten von Queries

Fragen und Diskussion