Visual Analytics für raum-zeitliche Daten

Meilenstein 2

Behdad Karimi

Maximilian Andres

GFZ Helmholtz Centre Potsdam

Humboldt-Universität zu Berlin

1.1 Gründe für den verwendeten math. Datentyp

Wir haben uns für den mathematischen Datentyp **Relation** entschieden.

Grund:

- Die Relation ist eine Menge aus n-Tupeln und ermöglicht die Ausführung von mathematischen Grundoperationen im Stile eines Datenbanksystems.
- Das Datenbanksystem fordert eine ordnende Struktur der Daten. Die geforderte ordnende Struktur wird durch unsere Daten ermöglicht.
- Der Timestamp dient dabei als ordnende Variable. Jeder Timestamp ist ein Small, Large, RelHumidity und OutdoorTemp Wert zugeordnet. Eine Reihe dieser fünf Variablen bildet ein Tupel.

GFZ Helmholtz Centre Potsdam

Humboldt-Universität zu Berlin

1.1 Gründe für den verwendeten math. Datentyp

Wir haben uns für den mathematischen Datentyp **Relation** entschieden.

Grund:

- Die Gesamtheit dieser Tupel bildet eine Relation. Auf dieser Relation können die Grundoperationen Selection, Projection und Aggregation ausgeführt werden.

GFZ Helmholtz Centre Potsdam

Humboldt-Universität zu Berlin

1.2 Gründe für die gewählte Implementierung

Die gewählte Implementierung besitzt zwei Klassen: Table und DataSource.

Die Gründe für die Implementierung der Klasse DataSource:

- Ermöglicht das Laden der gegebenen Staubdaten in eine Instanz der Klasse Table und das instanziieren einer Table mit modifizierten Staubdaten.
- Ermöglicht das Erstellen von unterschiedlichen Staubdaten im gleichen Rahmen.
- Anwendung der Funktionen Selection, Projection und Aggregation auf base_data und modifizierte Tabellen anwendbar

GFZ Helmholtz Centre Potsdam

Humboldt-Universität zu Berlin

1.2 Gründe für die gewählte Implementierung

Die gewählte Implementierung besitzt zwei Klassen: Table und DataSource.

Die Gründe für die Implementierung der Klasse Table:

- Der Fokus war eine standardisierte Relation der Staubdaten. Dabei haben wir besonders Acht auf die Implementierung des Timestamps als datetime gelegt. Da dadurch die Aggregation in der Range von datatime ermöglicht wird.

GFZ Helmholtz Centre Potsdam

Humboldt-Universität zu Berlin

2. ReadData

- liest Daten von der Festplatte oder Laufzeitdaten
- Erstellung einer Instanz der Klasse Table (in beiden Fällen)
- Table liest externe Datei in Dataframe oder übernimmt Laufzeitdaten
- base_data nicht vorhanden -> eingelesener Datensatz wird zu base_data
- table_count zählt nicht-base_data Instanzen der Klasse Table
- table_count >50 -> Löschen des ältesten nicht-base_data Datensatzes

3. Operatoren

- Anwendung auf alle Daten möglich (base_data und Zwischenergebnisse)
- Selection
  - same, at_least, less_than, span
  - Speicherung des Ergebnisses optional via Funktion readData
- Projection
  - Reduzierung auf zwei Spalten
  - Speicherung via readData
- Aggregation
  - min, max, avg
  - falls range gegeben: Verwendung von Selection ohne Speicherung
  - Speicherung via readData

GFZ Helmholtz Centre Potsdam

Humboldt-Universität zu Berlin

4. Abbildung der Laufzeit Verhaltens

	projection	selection	aggregation
10k Elemente	0.0005750	0.0014005	0.0013356
100k Elemente	0.0017588	0.0032707	0.0035138
1 Mio Elemente	0.0117051	0.0180345	0.0231888

GFZ Helmholtz Centre Potsdam

Humboldt-Universität zu Berlin

5. Abbildung der Laufzeit Verhaltens

	read time	projection	selection	aggregation
10k Elemente	0.023290	0.0005750	0.0014005	0.0013356
100k Elemente	0.264147	0.0017588	0.0032707	0.0035138
1 Mio Elemente	2.636066	0.0117051	0.0180345	0.0231888
10 Mio Elemente	24.025017	0.2135309	0.1601825	0.2761791
100 Mio Elemente	1017.017759	X	79.9390525	X

GFZ Helmholtz Centre Potsdam

Humboldt-Universität zu Berlin

6. Beweisskizze

- **Identify:** $\sigma_{Kondition}(A)$
- **Compare:** $(\pi_{Spalte1}(\sigma_{Kondition1}(A)) \times (\pi_{Spalte1}(\sigma_{Kondition2}(A)))$
- **Determine:** $\pi_{Spalte1}(\sigma_{Kondition}(A))$