Seminar Data Streams
Thema: Anfragesprachen

Sebastian Glöckner
s_gloeck@informatik.uni-kl.de

TECHNISCHE UNIVERSITÄT
KAISERSLAUTERN

Inhaltsübersicht

- Einleitung
- Grundlagen Datenstrom-orientierter Anfragesprachen
- Vorstellung einiger Anfragesprachen
  - CQL
  - Hancock
  - Aurora
  - GSQL
  - StreaQuel
- Vergleich der Anfragesprachen
- Vergleich der Anfragesprachen zu SQL
- Zusammenfassung

Einleitung

- Neue Anwendungen stellen neue Anforderungen an Datenverwaltungssysteme
  - Sensornetzwerke
  - Positionsüberwachung
  - Netzwerkanalyse
  - ...
- Andere Art der Verarbeitung und Speicherung gewünscht/erforderlich
- Herkömmliche DBMS sind dazu nicht geeignet
- Neue, spezielle Datenverwaltungssysteme: DSMS

Grundlagen Datenstrom-orientierter Anfragesprachen (1)

- Hohe Datenaufkommen, müssen schnell verarbeitet werden
- Keine persistente Speicherung aller Daten
- Erzeugung aggregierter Daten (als Anfrageergebnis)
- Approximierte Ergebnisse meist tolerierbar
- Problem: Eingabestrom potenziell unendlich
  - Blockierende Operatoren dadurch nicht möglich
  - Lösung: Fenstertechniken (Sliding Windows) □

Grundlagen Datenstrom-orientierter Anfragesprachen (2)

- Anfragen im System
  - Zu Beginn oder während Betrieb gestellt
  - Kontinuierlich bearbeitet oder nur einmal
- Relationen und Ströme als Verarbeitungstypen
- Einführung einer Zeitsemantik mit einer zeitlichen Ordnung (logische und physikalische Zeit)
- Im wesentlichen zwei Arten von Anfragesprachen
  - Deklarativ, z. B. CQL, StreaQuel, GSQL
  - Prozedural, z. B. Hancock

Vorstellung einiger Anfragesprachen: CQL(1)

- Continuous Query Language, stammt aus dem STREAM-Projekt der Stanford Universität
- Syntax und Fenstertechniken an SQL2003 angelehnt
- Unterstützt Relationen und Ströme
- Zeitdomäne mit geordneten Elementen
- Drei Klassen von Operatoren
  - Strom-zu-Relation
  - Relation-zu-Strom
  - Relation-zu-Relation

Vorstellung einiger Anfragesprachen: CQL(2)

- Operationen auf Relationen
- Strom-zu-Relation- und Relation-zu-Strom-Operatoren zur Konvertierung
- Relation-zu-Relation-Operatoren aus SQL
- Strom-zu-Relation durch Sliding Windows
  - Zeitbasiert
  - Tupelbasiert
  - Partitionsbasiert
- Relation-zu-Strom-Operatoren
  - Istream (insert stream)
  - Dstream (delete stream)
  - Rstream (relation stream)

Vorstellung einiger Anfragesprachen: CQL(3)

- Beispiel: Gegeben sei eine Anwendung, welche dazu dient den Verkehr auf einer Autobahn mittels Sensoren und Lichtschranken zu erfassen
- Strom: ABStrGeschw (Spur, Streckenabschnitt, speed)

```
Select Istream(*)
From ABStrGeschw [Range 2 hours]
Where speed > 100
```

- liefert einen Strom mit den Daten derjenigen Fahrzeuge, die in den letzten 2 Stunden gemessen wurden und schneller als 100 (km/h) fuhren.

Vorstellung einiger Anfragesprachen: Hancock(1)

- Von AT&T entwickelt zur Verarbeitung von Daten von Telekommunikationssystemen (z. B. Erfassung von Mobiltelefongesprächsdaten)
- Enorme Datenmengen, viele Redundanzen, exakte Speicherung nicht möglich
- Daten gewinnen erst durch Aggregation Relevanz
- Speicherung sogenannter Signaturen
- Früher Verwendung von C-Programmen
  - Schnell und effizient
  - Aber auch kompliziert und schwer wartbar

Vorstellung einiger Anfragesprachen: Hancock(2)

- Entwicklung einer eigenen, auf C basierenden, Anfragesprache
  - Spezielle Konstrukte
  - Vordefinierte Funktionen
  - Leicht anpassbar und erweiterbar
- Sliding Windows
- Events
  - Repräsentiert durch definierte Datenvorkommen
  - Erkennungsfunktionen, verwenden Fenster
  - Bei Signalisierung abarbeiten definierter Anweisungen
- Üblicher Verarbeitungsablauf
  - Erfassen von Daten über einen Zeitraum
  - Hauptmethode sichert Signaturen
  - Ruft Verarbeitungsprozedur auf
  - Diese verarbeitet Events und verändert die Signaturen

Vorstellung einiger Anfragesprachen: Hancock(3)

- Beispielprogramm:

```
Void out(AWS_s calls, cellTower_m ct){
   profile p;
   iterate    (
         over calls
         filteredby completeCellCall
         sortedby origin
         withevents originDetect )
   {
        event line_end(pn_t mpn) {
               profile mytemp;
               mytemp = ct<:mpn:>;
               ct<:mpn:> = update(mytemp,p);
        }
   };
}
```

Vorstellung einiger Anfragesprachen: Aurora(1)

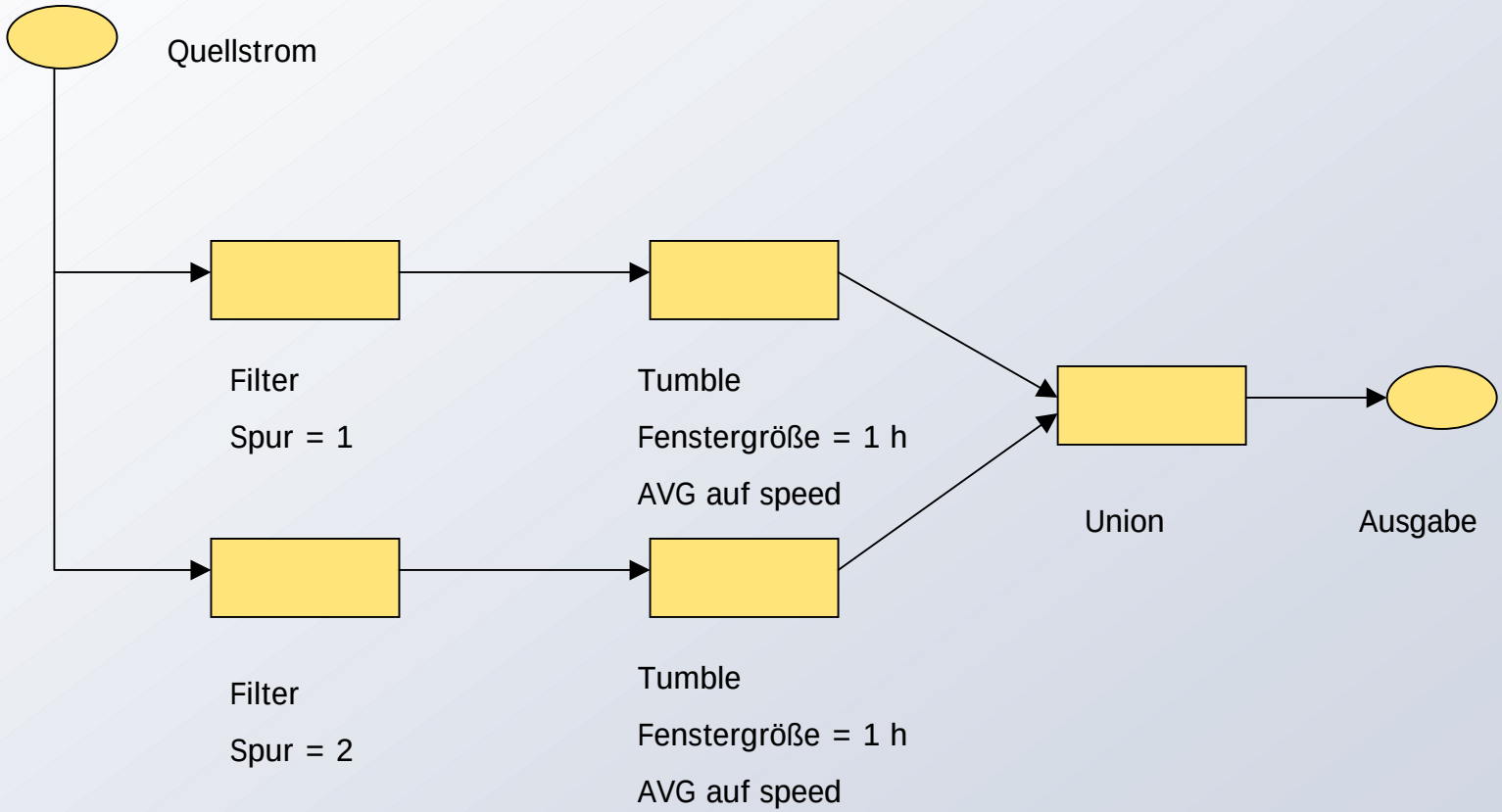
- DSMS von MIT, Brandeis und Brown University
- Im Gegensatz zu anderen Systemen Verwendung einer graphischen Anfragespezifikation (Grundlage: SQuAL)
- Anfragesprache besteht aus Boxen (Operatoren) und Pfeilen (Datenfluss)
- Anfrage ist ein Netzwerk aus Boxen und Pfeilen
- Connection Points
- Ad-Hoc Anfragen, kontinuierliche Anfragen und Views

Vorstellung einiger Anfragesprachen: Aurora(2)

- Operatoren nur auf Strömen
- Fenster-Operatoren definieren ein Fenster auf dem Datenstrom, welches durch eine Benutzerdefinierte Funktion verarbeitet wird
  - Slide
  - Tumble
  - Latch und Resample
- Weitere Operatoren
  - Filter und Drop
  - Join und Union
  - Map

Vorstellung einiger Anfragesprachen: Aurora(3)

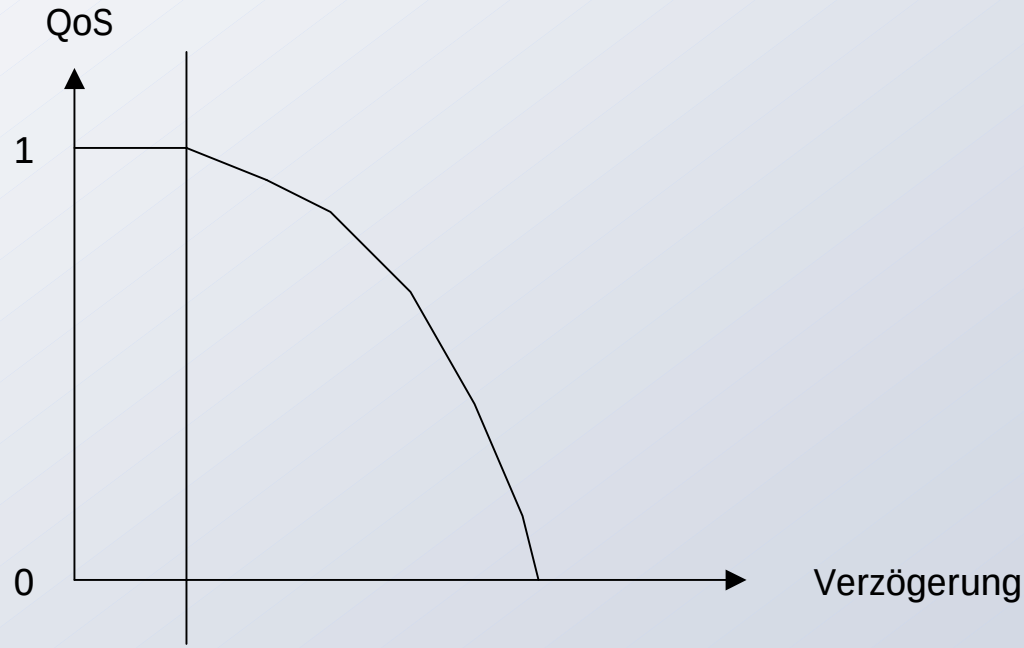
- Beispiel Anfrage:



Vorstellung einiger Anfragesprachen: Aurora(4)

- QoS-Spezifikation zu jeder Anfrage durch 3 Graphen
  - Verzögerungsgraph, obligatorisch
  - Auslassungsgraph
  - Wertegraph

- Beispiel QoS-Graph:



Vorstellung einiger Anfragesprachen: GSQL(1)

- Verwendet in Gigascope von AT&T, ein System zur Überwachung und Auswertung von Netzwerkdaten
- Angelehnt an SQL
- Operiert nur auf Strömen
- Anfragen können benannt und referenziert werden
- Fenster, zur Auflösung blockierender Operatoren, mit Hilfe von Attributen, auf denen Ordnungseigenschaften definiert wurden

Vorstellung einiger Anfragesprachen: GSQL(2)

- Ordnungseigenschaften werden i.d.R. auf Zeitattributen definiert (logische und physikalische Zeit), z. B.
  - Strikt/monoton zu-/abnehmend
- Beispiele einer Join-Bedingung auf zwei Strömen B und C für ein Attribut ts mit monoton zunehmender Ordnung:
  - B.ts = C.ts
  - B.ts >= C.ts - 1 und B.ts <= C.ts + 1
- Zur Zeit in GSQL implementiert: Selektion, binärer Join, Aggregation und Merge
- Benutzerdefinierte Funktionen, externe Parameter

Vorstellung einiger Anfragesprachen: GSQL(3)

- Beispielanfragen:

```
DEFINE {query name tcpDest0; }
Select destIP, destPort, time
From eth0.TCP
Where IPVersion = 4 and Protocol = 6

DEFINE {query name tcpDest; }
Merge tcpDest0.time : tcpDest1.time
From tcpDest0, tcpDest1
```

Vorstellung einiger Anfragesprachen: StreaQuel(1)

- SQL-ähnlich
- Nur Ströme
- große Bandbreite an Fenstervarianten
- Basis-Syntax:

```
SELECT projection_list
FROM   from_list
WHERE selection_and_join_predicates
ORDEREDBY
TRANSFORM…TO
WINDOW…BY
```

Vorstellung einiger Anfragesprachen: StreaQuel(2)

- Die Transform-Klausel beschreibt Fensterverlauf
  (ST=Startzeitpunkt)

```
Transform Stream1
For ( t = ST; t < ST + 10; t++) To Stream1(t)
Window Stream1 By ST, t
```

- Erzeugte Fenster wenn ST = 40
  [40,40],[40,41],[40,42],...,[40,49]

- Mögliche Fenstertypen
  - Sliding
  - Snapshot

Vorstellung einiger Anfragesprachen: StreaQuel(3)

- Beispiele:

```
Select Alert()
From ABStrGeschw ab
Where ab.Streckenabschnitt = 55
      AND ab.speed > 130
Having COUNT (*) > 20
```

```
Select Alert()
From ABStrGeschw ab
Where ab.Streckenabschnitt = 55
      AND ab.speed > 130
Having COUNT (*) > 20
```

```
Window ab by (NOW – 1h, NOW)
```

Vergleich der vorgestellten Anfragesprachen

	CQL	Hancock	Aurora	GSQL	StreaQuel
Fenster	Zeit,Tupel, Partition	Ja	Slide,Tumble etc.	Ordnungs-basiert	umfangreich
Zeitsemantik	Ja	Nein	Ja	Ja	Ja
Ströme + Relationen	Ja	Jein	Nein	Nein	Nein
Blockierende Operatoren	Mit Sliding Windows	Ja	Ja	Mit geordneten Attributen	Mit Sliding Windows
Approxi-mation	Nein	Ja	Ja	Nein	Nein
Persistenz	Nein	Ja	Temporär	Nein	Nein
QoS	Nein	Nein	Ja	Nein	Nein

Vergleich der vorgestellten Anfragesprachen mit SQL

	DBMS	DSMS
Fenstertechnologien	erst seit SQL2003	Ja
Ströme	Nein	Ja
Approximation	Nicht erforderlich	Teilweise erforderlich
Zeitsemantik	Nein	Ja
Blockierende Operatoren	Kein Problem	Problematisch
Persistenz	Elementarer Bestandteil	Nur selten
QoS	Keine Abstufung, „Full Service“	Teilweise notwendig
„Universalität“, Verbreitung	Ja, hoch	Nein, gering

Zusammenfassung

- Datenströme erfordern eine andere Art der Verarbeitung, und wenn überhaupt, der Speicherung
- DSMS bisher meist auf eine bestimmte Anwendung zugeschnitten, daher einige Unterschiede, z. B. bei Persistenz, Approximation und Fenstertechnologien
- Viele Elemente jedoch prinzipiell in fast jeder Sprache, z.B. Fenster, Zeitsemantik, Ströme
- Kaum universelle Ansätze, die alle Anforderungen voll erfüllen (STREAM und Aurora versuchen dies jedoch zumindest)
- QoS spielt bisher kaum eine Rolle