

Stream-Verwaltung mit konventionellen Datenbanksystemen

Christian Rensch

21.06.2005

Seminar Data Streams

1

Thema

- ◆ Verwaltung von Datenströmen in konventionellen Datenbanksystemen
 - optimiert auf Daten geringer Update-Rate
 - Transaktionskonzept
- ◆ Oracle, IBM DB2, Microsoft SQL Server

21.06.2005

Seminar Data Streams

2

Agenda

- ◆ Probleme konventioneller Systeme
- ◆ Bisherige Lösungen
- ◆ Ansätze zur Datenstromverwaltung in konventionellen DBMS
- ◆ Zusammenfassung

21.06.2005

Seminar Data Streams

3

Eigenschaften von Datenströmen

- ◆ Kontinuierliche Folge von Datensätzen mit
 - vielen unterschiedlichen Quellen
 - stark variierender Rate
 - unbekanntem Ende

21.06.2005

Seminar Data Streams

4

Probleme konventioneller DBMS

- ◆ Speicherplatz
- ◆ Hohe Insert-Lasten
- ◆ Echtzeitanforderung
- ◆ Approximation
- ◆ Ad-hoc-Anfragen vs. vordefinierte Anfragen
- ◆ Human-Active-DBMS-Passive vs. DBMS-Active-Human-Passive

21.06.2005

Seminar Data Streams

5

Bisherige Lösungen

- ◆ (bisher) kein generisches Datenstromverwaltungssystem
- ◆ anwendungsspezifische Lösungen:
ankommende Daten werden direkt vom Anwendungsprogramm verarbeitet

21.06.2005

Seminar Data Streams

6

Nachteile

- ◆ teure Herstellung der Software:
"maßgeschneiderte" Lösungen
- ◆ teure Wartung:
Software-Team statt DB-Administrator

21.06.2005

Seminar Data Streams

7

Stream-Verarbeitung (I)

- ◆ durch spezielle Systeme:
 - Aurora
 - TelegraphCQ
 - STREAM
- ◆ durch konventionelle DBMS:
 - Nutzen vorhandener Konzepte
 - Weiterentwicklung

21.06.2005

Seminar Data Streams

8

Stream-Verarbeitung (II)

- ◆ nutzbare Ansätze:
 - Trigger und Assertions
 - Materialisierte Sichten
 - Partitionierung
 - Sampling
 - Analytische Funktionen

21.06.2005

Seminar Data Streams

9

Trigger und Assertions (I)

- ◆ Trigger:
 - bereits in den meisten DBMS implementiert
 - realisieren kontinuierliche Anfragen
 - DBMS reagieren eigenständig auf Ereignisse → Aktive DBMS
- ◆ ABER:
 - arbeiten nur auf einer Basisrelation
 - zu viele Trigger verlangsamen System

21.06.2005

Seminar Data Streams

10

Trigger und Assertions (II)

- ◆ Assertions:
 - operieren über mehrere Basisrelationen
- ◆ ABER:
 - nur Konsistenztest, kein Ausführen von Aktionen
 - obwohl Teil von SQL in keinem DBMS implementiert

21.06.2005

Seminar Data Streams

11

Materialisierte Sichten (I)

- ◆ Daten der Sichten werden physisch gespeichert
- ◆ im Vergleich zu virtuellen Sichten:
 - wiederholter Zugriff erfolgt schneller, insbesondere bei komplexen Anfragen und bei verteilten Datenbanken
- ◆ ABER:
 - höherer Speicherplatzbedarf
 - Daten müssen aktuell gehalten werden

21.06.2005

Seminar Data Streams

12

Materialisierte Sichten (II)

- ◆ automatisches Aktualisieren vs. manuelles Aktualisieren
- ◆ vollständiges Aktualisieren vs. Änderungs-Logging

21.06.2005

Seminar Data Streams

13

Materialisierte Sichten (III)

- ◆ Oracle:
 - vor Version 8: SNAPSHOT
 - danach: MATERIALIZED VIEW
- ◆ IBM DB2:
 - MATERIALIZED QUERY TABLE
- ◆ Microsoft SQL Server:
 - INDEXED VIEWS mit folgenden Einschränkungen:
 - ◆ TOP, MIN, MAX und COUNT verboten
 - ◆ Verwendung anderer Views verboten
 - ◆ UNION, OUTER JOIN und Unterabfragen verboten

21.06.2005

Seminar Data Streams

14

Materialisierte Sichten (IV)

- ◆ Query-Rewriting:

Nutzung vorhandener materialisierter Sichten, um neue Anfragen schneller beantworten zu können

21.06.2005

Seminar Data Streams

15

Partitionierung (I)

- ◆ Tabellen können mehrere Partitionen besitzen
- ◆ Datensätze werden nach festlegbaren Kriterien in Partitionen einsortiert, z.B.
 - ◆ Partitionierung nach einem Attribut 'Datum' in Geschäftsjahre
 - ◆ Partitionierung nach einem Attribut 'Land'

21.06.2005

Seminar Data Streams

16

Partitionierung (II)

```
CREATE TABLE Gespraech (Datum DATE, KdNr NUMBER, Dauer NUMBER)
PARTITION BY RANGE (Datum)
(PARTITION vor_2004 VALUES
LESS THAN TO_DATE('01-JAN-2004', 'DD-MON-YYYY'),
PARTITION in_2004 VALUES
LESS THAN TO_DATE('01-JAN-2005', 'DD-MON-YYYY'),
PARTITION nach_2004 VALUES LESS THAN MAXVALUE) ;
```

Datum	KdNr	Dauer
2003-05-15	1	45
2003-07-21	2	215
2004-01-16	1	200
2004-10-16	3	20
2005-03-07	2	50

TABLE Gespraech

PARTITION "vor_2004"

PARTITION "in_2004"

PARTITION "nach_2004"

21.06.2005

Seminar Data Streams

17

Partitionierung (III)

- ◆ Anfragenoptimierung
- ◆ direkter Zugriff auf einzelne Partitionen möglich → Aggregatfunktionen über Partitionen schneller
- ◆ Sperrende Operationen sperren nur angefragte Partitionen

21.06.2005

Seminar Data Streams

18

Partitionierung (IV)

- BY RANGE : Wertebereiche des Typs "von - bis"
- BY LIST : Nach einzelnen Werten
- BY HASH : über Hash-Funktion

◆ Oracle:

- BY RANGE
- BY LIST
- BY HASH

◆ IBM DB2:

- BY RANGE
- BY HASH

◆ MS SQL:

- keine Implementierung

Partitionierung noch nicht Teil des SQL-Standards!

Sampling (I)

◆ Idee:

zufällige Auswahl von Datensätzen
→ Approximation möglich

Anzahl ausgewählter Sätze abhängig
von der Sampling-Rate

Sampling (II)

- ◆ kein SQL-Standard-Konstrukt
- ◆ Umsetzung nur in Oracle:

```
SELECT AVG(Dauer) FROM Gespraech
```

nimmt alle Datensätze zur Berechnung des Durchschnitts

Sampling (II)

- ◆ kein SQL-Standard-Konstrukt
- ◆ Umsetzung nur in Oracle:

```
SELECT AVG(Dauer) FROM Gespraech  
SAMPLE (12)
```

benutzt jeden Datensatz mit einer Wahrscheinlichkeit von 12%

Sampling (II)

- ◆ kein SQL-Standard-Konstrukt
- ◆ Umsetzung nur in Oracle:

```
SELECT AVG(Dauer) FROM Gespraech  
SAMPLE (12) SEED (4)
```

sorgt für gleiche Auswahl bei erneuter
Anfrage

Analytische Funktionen (I)

auch: Window-Funktionen

- ◆ ermöglichen komplexe Anfragen
- ◆ Aggregatfunktionen
- ◆ vergleichbar mit GROUP BY
- ◆ NEU:
 - Resultset bleibt unverändert
 - mehrere Bereiche in einer Anfrage
 - Anfragen über Windows
(=gleitende Fenster innerhalb der Bereiche)

Analytische Funktionen (II)

Datum	KdNr	Dauer
2005-05-15	1	45
2005-05-15	2	215
2005-05-16	1	200
2005-05-16	2	125
2005-05-16	3	20
2005-05-17	2	50
2005-05-18	2	310

```
SELECT KdNr, SUM(*) KdSum
FROM Gespraech
GROUP BY KdNr
ORDER BY KdNr
```

KdNr	KdSum
1	245
2	700
3	20

21.06.2005

Seminar Data Streams

25

Analytische Funktionen (III)

Datum	KdNr	Dauer
2005-05-15	1	45
2005-05-15	2	215
2005-05-16	1	200
2005-05-16	2	125
2005-05-16	3	20
2005-05-17	2	50
2005-05-18	2	310

```
SELECT Datum, KdNr,
SUM(Dauer) OVER
(PARTITION BY KdNr) KdSum
FROM Gespraech
ORDER BY Datum, KdNr
```

Datum	KdNr	KdSum
2005-05-15	1	245
2005-05-15	2	700
2005-05-16	1	245
2005-05-16	2	700
2005-05-16	3	20
2005-05-17	2	700
2005-05-18	2	700

21.06.2005

Seminar Data Streams

26

Analytische Funktionen (IV)

Datum	KdNr	Dauer
2005-05-15	1	45
2005-05-15	2	215
2005-05-16	1	200
2005-05-16	2	125
2005-05-16	3	20
2005-05-17	2	50
2005-05-18	2	310

```
SELECT Datum, KdNr,
SUM(Dauer) OVER () GesSum
FROM Gespraech
ORDER BY Datum
```

Datum	KdNr	GesSum
2005-05-15	1	965
2005-05-15	2	965
2005-05-16	1	965
2005-05-16	2	965
2005-05-16	3	965
2005-05-17	2	965
2005-05-18	2	965

21.06.2005

Seminar Data Streams

27

Analytische Funktionen (V)

Datum	KdNr	Dauer
2005-05-15	1	45
2005-05-15	2	215
2005-05-16	1	200
2005-05-16	2	125
2005-05-16	3	20
2005-05-17	2	50
2005-05-18	2	310

```
SELECT Datum, KdNr,
SUM(Dauer) OVER
(PARTITION BY KdNr) KdSum,
SUM(Dauer) OVER
(PARTITION BY Datum) DatSum
FROM Gespraech ORDER BY Datum, KdNr
```

Datum	KdNr	KdSum	DatSum
2005-05-15	1	245	260
2005-05-15	2	700	260
2005-05-16	1	245	345
2005-05-16	2	700	345
2005-05-16	3	20	345
2005-05-17	2	700	50
2005-05-18	2	700	310

21.06.2005

Seminar Data Streams

28

Analytische Funktionen (VI)

Datum	KdNr	Dauer
2005-05-15	1	45
2005-05-15	2	215
2005-05-16	1	200
2005-05-16	2	125
2005-05-16	3	20
2005-05-17	2	50
2005-05-18	2	310

```
SELECT Datum, KdNr,  
SUM(Dauer) OVER  
(PARTITION BY KdNr ORDER BY Datum  
RANGE BETWEEN 1 DAY PRECEDING  
AND CURRENT ROW) SumLetzte2  
FROM Tabelle ORDER BY Datum, KdNr
```

Datum	KdNr	SumLetzte2
2005-05-15	1	45
2005-05-15	2	215
2005-05-16	1	245
2005-05-16	2	340
2005-05-16	3	20
2005-05-17	2	175
2005-05-18	2	360

21.06.2005

Seminar Data Streams

29

Analytische Funktionen (VII)

◆ Noch mehr?

- ◆ alle bekannten Aggregatfunktionen
- ◆ ROW_NUMBER, RANK und DENSE_RANK nummerieren Reihen eines Bereiches durch
- ◆ FIRST_VALUE (Attribut) und LAST_VALUE (Attribut) liefern den ersten bzw. den letzten Wert eines Attributs in einem Bereich
- ◆ LAG (Attribut, n, Default) und LEAD (Attribut, n, Default) liefern den Wert eines Attributs n Zeilen vor bzw. nach der betrachteten Zeile – falls diese nicht existiert gibt sie Default zurück

21.06.2005

Seminar Data Streams

30

Zusammenfassung (I)

- ◆ Trigger und Assertions
 - ◆ Aktive DBMS → Selektion von Daten
 - ◆ kontinuierliche Anfragen
- ◆ Materialisierte Sichten
 - ◆ Schnellere Abfragemauswertung → Echtzeit
- ◆ Partitionierung
 - ◆ Selektion
 - ◆ Echtzeit
 - ◆ Approximation
- ◆ Sampling
 - ◆ Approximation
- ◆ Analytische Funktionen
 - ◆ Echtzeit
 - ◆ Komplexe Analyseabfragen

21.06.2005

Seminar Data Streams

31

Zusammenfassung (II)

- ◆ durch vorgestellte Konzepte einzelne Aspekte nachbildbar
- ◆ ABER:
 - Problem bleibt "PUSH" von Änderungen
 - alle Nachbildungen mangelt es an Leistung, insbesondere hohe Insert-Last schwer zu verarbeiten

21.06.2005

Seminar Data Streams

32

Weitere Informationen

◆ Allgemein:

- ◆ K. E. Kline:
SQL in a Nutshell, 2nd Edition
- ◆ <http://www.craigsmullins.com/>
- ◆ D. Quass, A. Gupta, I.S. Mumick, J. Widom:
Making Views Self-Maintainable for Data Warehousing
- ◆ Lukasz Golab:
Querying Sliding Windows over On-Line Data Streams

21.06.2005

Seminar Data Streams

33

Weitere Informationen

◆ Oracle:

- ◆ <http://www.databasejournal.com/features/oracle/article.php/2192071>
- ◆ http://www.ordix.de/oneWS2/2_2002/siteengine/artikel/sundd_2.html
- ◆ <http://www.orafaq.com/articles/archives/000060.htm>
- ◆ <http://www.psoug.org/reference/partitions.html>

◆ IBM DB2:

- ◆ <http://publib.boulder.ibm.com/infocenter/dzichelp/index.jsp>
- ◆ <http://www.db2mag.com/story/showArticle.jhtml?articleID=17701065>

◆ Microsoft SQL Server:

- ◆ <http://www.sqlteam.com/item.asp?ItemID=1015>
- ◆ <http://msdn.microsoft.com/>
- ◆ <http://www.hansevision.de/downloads/SQLServer2005.ppt>

21.06.2005

Seminar Data Streams

34