

Strumieniowe bazy danych

STREAM: The Stanford Data Stream Management System

Michał Stochmiałek

`<misto@e-informatyka.pl>`

Plan prezentacji

Wprowadzenie

- Problem na przykładzie
- Systemy DSMS
- Projekt STREAM

STREAM i CQL

- Podstawowe założenia
- Język CQL
- Przykład

Podsumowanie

Problem

System analizujący ruch sieciowy:

- ▶ analiza wszystkich pakietów w wielu podsieciach, (*nieprzerwany strumień danych z wielu źródeł*)
- ▶ podczas analizy wykorzystywany:
 - ▶ zbiór znanych użytkowników (identyfikacja poprzez adresy IP)
 - ▶ dane historyczne, poprzednie analizy
- ▶ statystyki
 - ▶ lista **k** najbardziej obciążających sieć adresów IP w ostatnich 20min
 - ▶ procentowe wykorzystanie złącza przez klientów w ostatniej godzinie
 - ▶ analiza sekwencji pakietów
 - ▶ analizy *ad hoc*

Jakie rozwiązanie zastosować?

- ▶ zastosowanie DBMS
 - ▶ czy ich konstrukcja przewiduje przetwarzanie takiej ilości danych?
- ▶ rozwiązania połowiczne
- ▶ rozwiązania dedykowane

Strumienie danych

- ▶ ciągły, nieograniczony, zmienny w czasie, dynamiczny strumień elementów danych
- ▶ przykłady systemów:
 - ▶ monitorowanie i analiza ruchu sieciowego, czy drogowego
 - ▶ logowanie dostępu do serwisów, logowanie kliknięć na stronie
 - ▶ zapis połączeń telefonicznych
 - ▶ analiza finansowa

DBMS a DSMS

Data Base Management System

- ▶ trwałe (*persistent*) relacje
- ▶ zapytania wykonywane w stosunkowo krótkim czasie
- ▶ swobodny dostęp (*random access*)
- ▶ stabilny plan zapytań

Data Stream Management System

- ▶ ulotne (*transient*) strumienie (i trwałe relacje)
- ▶ ciągłe zapytania (*continuous queries*)
- ▶ dostęp sekwencyjny
- ▶ nieprzewidywalna charakterystyka danych

STREAM: The Stanford Data Stream Management System

- ▶ projekt rozwijany na Uniwersytecie w Stanford
- ▶ duża ilość publikacji naukowych dotyczących różnych aspektów DSMS
- ▶ źródła oraz demonstracja na stronie projektu

Inne projekty

- ▶ Niagara (University of Wisconsin-Madison, the Oregon Health & Science University)
- ▶ Aurora (Brandeis University, Brown University, and MIT)
- ▶ Telegraph (UC Berkeley's Computer Science Division)
- ▶ MonaCQ (Instytut Techniki i Aparatury Medycznej, Zabrze)

STREAM i CQL

Podstawowe założenia

- ▶ czas spełnia bardzo ważną rolę w systemach DSMS
- ▶ czas jest dyskretny i składa się z *chwil*

$$\tau \in \mathcal{T}, \quad \mathcal{T} = \{1, 2, \dots\}$$

- ▶ metafora *uderzeń serca*

Typy danych

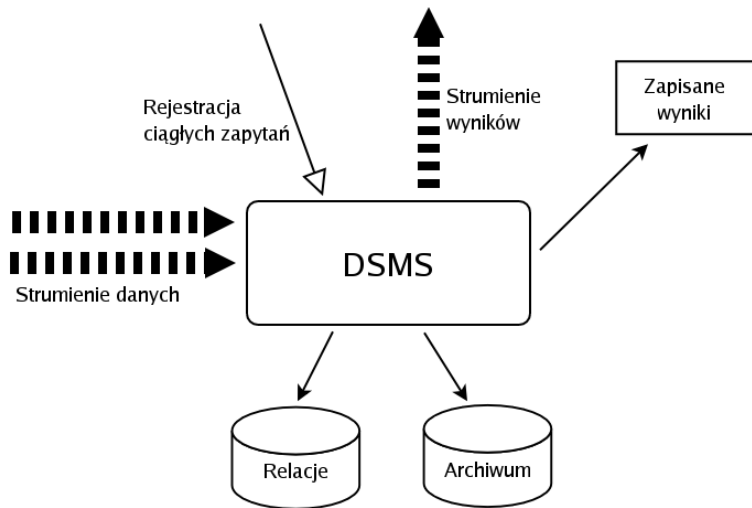
- ▶ **strumień** - sekwencja uporządkowana czasem. Każdy element strumienia składa z krotki danych oraz czasu pojawienia się jej w strumieniu (*timestamp*)

$$S = \{ \langle s, \tau \rangle : s - \text{krotka}, \tau \in \mathcal{T} \}$$

- ▶ **relacja** - tradycyjna relacja bazodanowa, ale z *poczuciem czasu*. Funkcja odwzorująca czas na zbiór krotek w danej chwili τ .

$$R(\tau) - \text{relacja } R \text{ w chwili } \tau$$

Ogólnikowy schemat działania DSMS



Język zapytań CQL

- ▶ CQL bazuje na języku SQL
- ▶ dodaje pojęcie i interpretację strumienia
- ▶ dodaje operacje na strumieniach
- ▶ dodaje metody konwersji pomiędzy strumieniami i relacjami
 - ▶ operatory strumień-relacja
 - ▶ operatory relacja-strumień

Operatory strumień-relacja

- ▶ koncepcja przesuwającego się okna (ang. *sliding window*)
- ▶ ekstrakcja najnowszych danych ze strumienia do postaci relacji
- ▶ okno oparte na czasie
S [Range 40 Seconds], S [Now]
- ▶ okno oparte na ilości krotek
S [Rows 40]

Operatory relacja-relacja

- ▶ język CQL dziedziczy z języka SQL wszystkie operatory relacyjne

```
SELECT P.pracownik_id, P.imie, P.nazwisko
FROM wchodzacyDoBudynkuStream [Range 30 Seconds] as W,
     pracownicy as P
WHERE W.pracownik_id = P.pracownik_id
```

Operatory relacja-strumień

- ▶ **strumień elementów dodanych** $IStream$ - w każdej chwili do strumienia dodawane są krotki, które zostały dodane w tej samej chwili do relacji R
- ▶ **strumień elementów usuniętych** $DStream$ - w każdej chwili do strumienia dodawane są krotki usunięte w tej samej chwili z relacji R
- ▶ **strumień elementów relacji** $RStream$ - w każdej chwili τ do strumienia dodawane są **wszystkie** elementy relacji R

Przykład: System naliczający opłaty za autostradę

- ▶ każdy samochód na autostradzie co 30s wysyła informację o swojej pozycji i prędkości
- ▶ autostrada podzielona jest na odcinki
- ▶ opłata za przejazd jest naliczana tylko gdy dany odcinek jest zatłoczony
- ▶ odcinek jest zatłoczony, gdy od 5min samochody poruszają się z prędkością niższą niż 60 km/h.

Przykład: Schemat strumienia danych

`predkoscSamochodowStream(samochodId, odcinekId, predkosc)`

- ▶ `samochodId` - identyfikator samochodu
- ▶ `odcinekId` - identyfikator odcinka autostrady
- ▶ `predkosc` - aktualna prędkość samochodu

Zapytanie 1: zbiór zatłoczonych odcinków autostrady

- ▶ odcinek jest zatłoczony, gdy samochody od 5min poruszają się z prędkością niższą niż 60 km/h.
- ▶ wynik: relacja zatłoczoneOdcinki(odcinekId)

```
SELECT odcinekId
FROM predkoscSamochodowStream [Range 5 Minutes]
GROUP BY odcinekId
HAVING AVG(predkosc) < 60
```

Zapytanie 2: liczba samochodów na autostradzie

- ▶ wysokość opłaty jest zależna od liczby samochodów na zatłoczonym odcinku
- ▶ samochód opuścił autostradę, jeżeli od 30s nie podał swojej pozycji
- ▶ wynik: relacja liczbaSamochodow(odcinekId, liczba)

```
SELECT odcinekId, COUNT(samochodId) as liczba  
FROM predkoscSamochodowStream [Range 30 Seconds]  
GROUP BY odcinekId
```

Zapytanie 3: wysokość opłaty za autostradę

- ▶ wysokość opłaty obliczamy według wzoru:

$$\textit{bazowaOpłata} \times (\textit{ilośćSamochodów} - 150)$$


- ▶ wynik: strumień opłaty(*samochodId*, *opłata*)

```
SELECT RStream(S.samochodId,  
              oplataBazowa * (LS.liczba - 150) as oplata  
FROM   predkoscSamochodowStream [Now] as S,  
       zatloczoneOdcinki as ZO, liczbaSamochodow as LS  
WHERE  S.odcinekId = ZO.odcinekId  
       and ZO.odcinekId = LS.odcinekId
```

Podsumowanie

- ▶ **strumieniowe bazy danych** - nowa klasa baz danych
- ▶ **strumień danych** - ciągły, dynamiczny, zmienny w czasie napływ nowych danych
- ▶ **ciągłe zapytania** - zapytania wykonywane często przez cały okres działania aplikacji opartej na DSMS

Źródła

-  B. Babcock, S. Babu, M. Datar, R. Motwani, J. Widom, *Models and Issues in Data Stream Systems*, czerwiec 2002
-  The STREAM Group, *Stanford Data Stream Management System*, marzec 2003
-  A. Arasu, S. Babu, J. Widom, *The CQL Continuous Query Language: Semantic Foundations and Query Execution*