Zastosowanie strumieniowych baz danych

Krzysztof Gogól

Agenda

1. Dlaczego strumieniowe bazy danych?
2. Prototypowe DSMS
3. Otwarte problemy
4. DSMS czyli okna
5. Przykłady zastosowania

Agenda

1. Dlaczego strumieniowe bazy danych?
2. Prototypowe DSMS
3. Otwarte problemy
4. DSMS czyli okna
5. Przykłady zastosowania

Dlaczego strumieniowe bazy danych?

Definicja

Bazy danych, przetwarzające ciągle i gwałtownie napływające informacje, przyjęto określać terminem *strumieniowych baz danych.*

Wysoko poziomowy podział:

5. Transakcyjne strumienie danych
6. Pomiarowe strumienie danych

Dlaczego strumieniowe bazy danych?

Przykładowe case-study - System czujników raportujących natężenie ruchu na autostradzie

Zbierane informacje:

- Identyfikator samochodu i jego prędkość
- Id autostrady i odcinka, na którym znajduję się samochód

Przykładowe zastosowanie:

A. Pomiar średniej prędkości samochodów na odcinku autostrady => informowanie nadjeżdżających kierowców o natężeniu ruchu

B. Wykrywanie wypadków (nagłe zatrzymanie samochodu)

C. Zarządzania sygnalizacją świetlną

Dlaczego strumieniowe bazy danych?

Dla większość problemów z dziedziny *data miningu* zostały opracowane efektywne algorytmy. Ich łatwe przeniesienie do *stream miningu* nie jest niestety możliwe.

O niezwykłości problemów stream miningu decydują:

- Brak możliwości wielokrotnego przeglądania zawartości bazy danych (*no multiple passes*).
- Zmienność danych w czasie (*Temporal locality*).
- Rozproszenie strumieni danych

Dlaczego strumieniowe bazy danych?

Tradycyjne DBMS a aplikacje strumieniowe:

- Jedno wykonanie (one-time) vs „ciągłość trwania” kwerend
- Czas
- Zmienność danych
- Zawodność danych
- Automatyczna reakcja

Dlaczego strumieniowe bazy danych?

Strumieniowe modele danych i języki zapytań

Timestamp

- **Implicit timestamp**
(numer porządkowy nadawany przez system zarządzający bazą danych)
- **Explicit timestamp**
(własność informacji, często związana z rzeczywistym czasem)

Okna

-**fizyczne** (oparte na czasie)

-**logiczne** (oparte na rozmiarze danych)

-**landmark** (ustalony jest jeden koniec przedziału)

-**sliding** (przesuwają się końce przedziału, nie zmienia się rozmiar okna)

Dlaczego strumieniowe bazy danych?

Przykładowe case-study - System czujników raportujących natężenie ruchu na autostradzie

```
SELECT exp_way, dir, seg, AVG(speed)
FROM CarSegStr [RANGE 5 MINUTES]
GROUP BY exp_way, dir, seg
```

Dlaczego strumieniowe bazy danych?

Ograniczenia implementacyjne

-operacje bezstanowe

-operacje nie blokujące

-działanie w czasie rzeczywistym

Przykładowe metryki oceny jakości DSMS:

- Czas odpowiedzi
- Dokładność
- Skalowalność

Agenda

1 Dlaczego strumieniowe bazy danych?

2 Prototypowe DSMS

3 Otwarte problemy

4 DSMS czyli okna

5 Przykłady zastosowania

Prototypowe DSMS

1. Aurora
2. COUGAR
3. Gigascope
4. Hancock
5. NiagaraCQ
6. StatStream
7. STREAM
8. TelegraphCQ
9. Tapestry

Agenda

1. Dlaczego strumieniowe bazy danych?
2. Prototypowe DSMS
3. Otwarte problemy
4. DSMS czyli okna
5. Przykłady zastosowania

Otwarte problemy

1. Data Stream Clustering

Bardzo znany problem w data miningu. Przeniesienie algorytmów do stream miningu jest trudne ze względu na ogranicznie one-pass. Ciekawe efekty uzyskuje zmodyfikowany algorytm k-średnich

2. Data Stream Classification

Główny problem stanowi ograniczenie temporal locality. Nie jest możliwe wykorzystanie algorytmów z data miningu.

3. Frequent Pattern Mining

Możliwe dwa podjeścia do problemu:

-ograniczenie badania do przesuwającego się okna

-badanie całego strumienia

Otwarte problemy

4. Change Detection in Data Streams

Problem podobny do wyszukiwania wzorca w strumieniu. Algorytmy wykrywające zmiany są szeroko stosowane w pomiarowych strumieniach danych

5. Stream Cube Analysis of Multi-dimensional Streams

Konieczność redukcji (agregacji) do mniejszej liczby wymiarów. Wykonywanie funkcji agregujących.

6. Loadshedding in Data Streams

DSMS nie ma wpływu na natężenie informacji przesyłanych strumieniami. Nawet w przypadku nagłego zwiększenia obciążenia system musi wykonywać kwerendy w zadanym czasie.

Otwarte problemy

7. Sliding Window Computation in Data Streams

Bardzo często w obliczeniach większą wagę mają nowsze dane. Starsze często nie ulegają już zmianie. Fakty te mogą zostać wykorzystane w algorytmach opierających się na przesuwających oknach.

8. Synopsis Construction in Data Streams

Wykorzystanie specjalnych struktur danych do przyspieszenia obliczeń (często przybliżonych).

9. Join Processing in Data Streams

Join jest fundamentalnym operatorem w bazach danych. Konieczne może być wykonywanie złączeń w ramach jednego strumienia lub połączenie kilku strumieni.

Otwarte problemy

10. Indexing Data Streams

Próby stworzenia systemu indeksowania, który w efektywny sposób przyspieszy działanie pewnych kwerend, funkcji agregujących, etc.

11. Dimensionality Reduction and Forecasting in Data Streams

Poszukiwanie korelacji pomiędzy strumieniami oraz próby przewidzenia dalszego zachowania się strumienia.

12. Distributed Mining of Data Streams

Problemy optymalizacji kosztów komunikacji, pamięci, informacji przechowywanych w węzłach

13. Stream Mining in Sensor Networks

Przetwarzanie ogromnych ilości informacji (przy ograniczonej pamięci, mocy obliczeniowej, etc)

Agenda

1. Dlaczego strumieniowe bazy danych?
2. Prototypowe DSMS
3. Otwarte problemy
4. DSMS czyli okna
5. Przykłady zastosowania

DSMS czyli okna

THE SLIDING WINDOW MODEL

Motywacja:
Nowsze dane mają większą wagę od starszych danych

Założenia:
Dane przypływają w sposób ciągły
Każda informacja traci ważność po N jednostkach czasu
Dla wykonywanych obliczeń istotne jest N ostatnich informacji

Cel:
Opracowanie algorytmów do wykonywania obliczeń statystycznych na danych wykorzystującego pamięć podliniową i okno rozmiaru N

ROAD MAP

Problem 1 (BASICCOUNTING)

Dany jest strumień danych, składający się z 0 i 1. Podawać w każdej chwili liczbę 1 w ostatnich N elementach.

Problem 2 (SUM)

Dany jest strumień danych, składających się z nieujemnych liczb naturalnych z zakresu [0, …, R]. Podawać w każdej chwili sumę ostatnich N elementów.

Wykorzystywana metoda:

Eksponencjalnych Histogramów (EH)

BASICCOUNTING

Trywialne rozwiązanie problemu (złożoność O(N)) jest rozwiązaniem optymalnym.

Dopuszczenie przybliżonych rozwiązań:

Algorytmy aproksymacyjne działają w czasie

O(log^ N /E)

Gdzie E jest dopuszczalnym błędem

O(log^ N /E) jest również dolnym ograniczeniem złożoności pamięciowej algorytmów aproksymacyjnych dla BasicCounting.

Dopuszczalność rozwiązań przybliżonych dla strumieniowych baz danych:

- Badanie ruchu w Internecie
- Szacowanie pamięci potrzebnej do wykonania operacji join

DSMS czyli okna

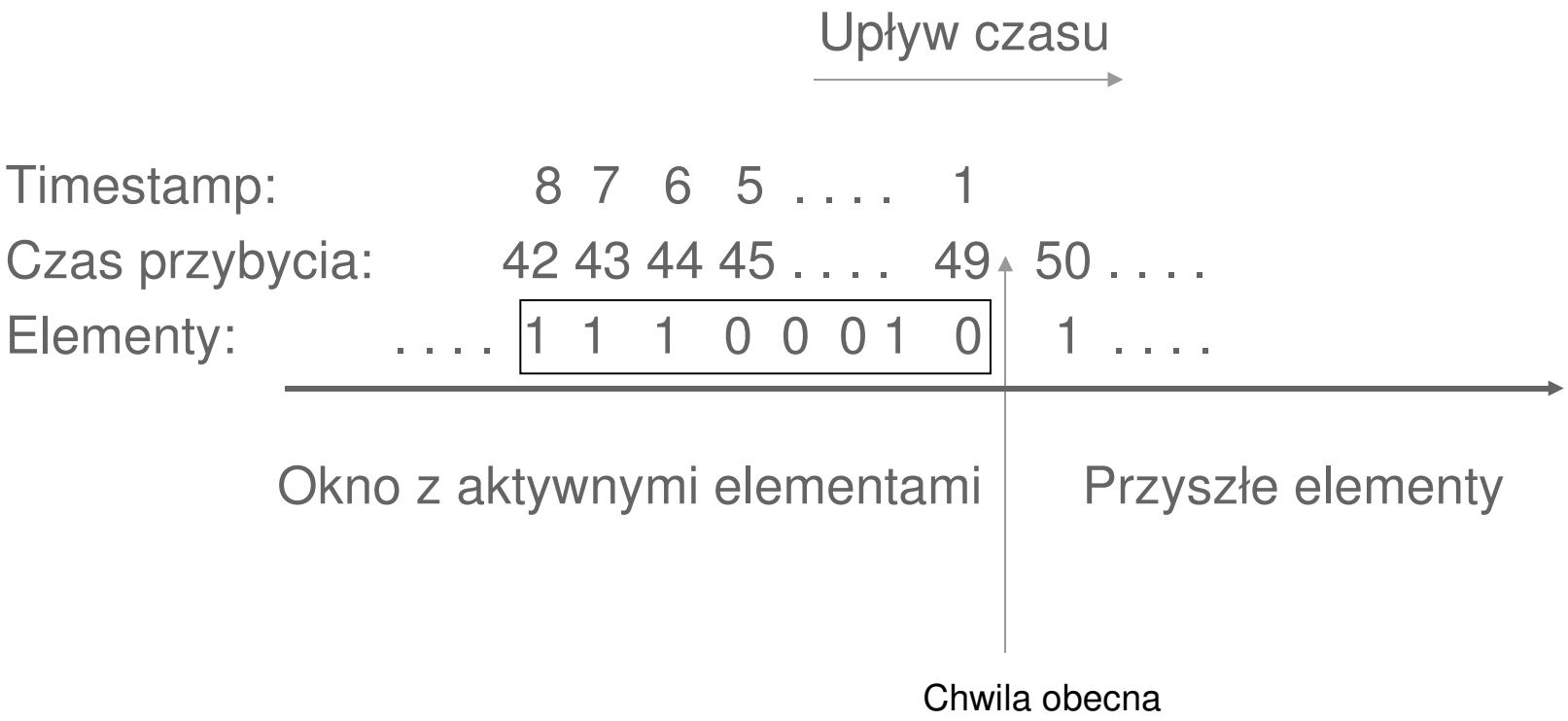
Rozwiązanie aproksymacyjne problemu BasicCounting:

1. Rozwiązanie intuicyjne
- Losowe próbkowanie = > Słaba dokładność

2. Rozwiązanie oparte na podziale okna na mniejsze cześci
- Podział na k równych części => Duże różnice w liczbie w 1 w poszczególnych częściach => Słaba dokładność
- Inny podział (metoda EH)

DSMS czyli okna

Oznaczenia:



Cześć z timestamp 2 i rozmiaru 2 oznacza dwie najbliższe 1s jedynki

U nas jedynki z chwil 2s i 6s

DSMS czyli okna

Algorytm (Insert)

2. Kiedy przybywa nowy element, policz nowe czasy życia. Jeśli czas życia ostatniej części przekracza N, usuń i zaktualizuj licznik LAST, zawierający rozmiar ostatniej części, oraz licznik TOTAL, zawierający łączną liczbę części okna.
3. Jeśli nowym elementem jest 0, nie rób nic więcej. W przeciwnym przypadku, utwórz nową część o rozmiarze 1 i zwiększ licznik TOTAL.
4. Przejrzyj listę części w kolejności rosnących rozmiarów. Jeśli jest k/2 + 2 części tego samego rozmiaru (k + 2 dla rozmiaru 1), scal dwie ostatnie części w jedną dwukrotnie większą. Uaktualnij licznik LAST, o ile jest to konieczne.

$k = [\ 1/E\ ]$

DSMS czyli okna

Przykład
Załóżmy, że k/2 = 1
Za każdym razem nowym elementem jest 1

32, 32, 16, 8, 8, 4, 4, 2, 1, 1	
32, 32, 16, 8, 8, 4, 4, 2, 1, 1, 1	(przybywa 1)
32, 32, 16, 8, 8, 4, 4, 2, 1, 1, 1, 1	(przybywa 1)
32, 32, 16, 8, 8, 4, 4, 2, 2, 1, 1	(scalamy 1)
32, 32, 16, 8, 8, 4, 4, 2, 2, 1, 1, 1	(przybywa 1)
32, 32, 16, 8, 8, 4, 4, 2, 2, 1, 1, 1, 1	(przybywa 1)
32, 32, 16, 8, 8, 4, 4, 2, 2, 2, 1, 1	(scalamy 1)
32, 32, 16, 8, 8, 4, 4, 4, 2, 1, 1	(scalamy 2)
32, 32, 16, 8, 8, 8, 4, 2, 1, 1	(scalamy 4)
32, 32, 16, 16, 4, 2, 1, 1	(scalamy 8)

DSMS czyli okna

Co z innymi algorytmami randomizowanymi:

- algorytmy Las Vegas,
- algorytmy Monte Carlo?

Dla jakich funkcji można stosować metodę HR?

Skąd nazwa metody?

Agenda

1. Dlaczego strumieniowe bazy danych?
2. Prototypowe DSMS
3. Otwarte problemy
4. DSMS czyli okna
5. Przykłady zastosowania

Gdzie można wykorzystać strumieniowe bazy danych?

Umiejętności dopotąd są jeszcze próżnym wynalazkiem, może czczym tylko rozumu wywodem, albo próżniactwa zabawą, dopokąd nie są zastosowane do użytku narodów.
I uczeni potąd nie odpowiadają swemu powołaniu, swemu w towarzystwach ludzkich przeznaczeniu, dopokąd ich umiejętność nie nadaje fabrykom i rękodziełom oświecenia, ułatwienia kierunku postępu.

Stanisław Staszic (1755–1826)

Przykład biznesowego zastosowania

Giełda Papierów Wartościowych w Warszawie

Potencjalne zadania:

-Obliczanie kursu

-Przewidywanie kursu

-Obliczanie funkcji statystycznych

-Testowanie hipotezy o zwiększeniu efektywności obliczeń (w oparciu o metody statystyki matematycznej)

-Próba zdefiniowania uniwersalnego współczynnika określającego konieczność zastąpienia relacyjnej bazy danych strumieniową

Przykład biznesowego zastosowania

A dokładniej:

1. Wyznaczanie kursu równowagi
2. Wyznaczanie kursu jednolitego
   - Algorytm 1 – średnia ważona
   - Algorytm 2 – w zależności od kursu odniesienia
   - Algorytm 3 – w zależności od liczby niezrealizowanych zleceń
3. Obliczanie wartości indeksów giełdowych
4. Algorytmy klastrowania vs podział na indeksy
4. Wycena instrumentów pochodnych (opcji)

Dziękuje za uwagę!