Poznań, 17.08.2012

dr hab. inż. Robert Wrembel, prof. nadzw.
Politechnika Poznańska
Wydział Informatyki
Instytut Informatyki
ul. Piotrowo 2
60-965 Poznań
tel. 61 665 2991, 600 974782
e-mail: Robert.Wrembel@cs.put.poznan.pl



# Recenzja rozprawy doktorskiej

## mgra Aleksandra Chrószcza

## Model przetwarzania strumieniowego uwzględniający zarówno synchronizację jak i język zapytań łączący paradygmaty języka obiektowego i deklaratywnego

**promotor: dr hab. inż. Marcin Gorawski, prof. nadzw.**

## 1. Tematyka rozprawy

Recenzowana rozprawa doktorska mgra Aleksandra Chrószcza w ogólności wpisuje się w problematykę zarządzania tzw. strumieniami danych (danymi strumieniowymi). Dane tego typu charakteryzują się tym, że napływają do systemu w sposób ciągły. Jako przykłady strumieni danych można wymienić m.in. dane z sensorów (np. instalacje przesyłu mediów, inteligentne budynki), notowania giełdowe, dane generowane przez zaawansowane infrastruktury transportu publicznego, dane monitoringu bazującego na RFID. Jednymi z podstawowych problemów badawczych i technologicznych w ww. zakresie są: analiza na bieżąco napływających danych, optymalizacja zapytań na strumieniach, składowanie strumieni. Tego typu problematyka jest podejmowana w ramach technologii Complex Event Processing, strumieniowych baz danych (stream databases), czy hurtowni danych czasu rzeczywistego (real-time data warehouses).

Problem analizy na bieżąco napływających danych wiąże się z opracowaniem języka zapytań umożliwiającego m.in. operowanie na wielu strumieniach danych i wyznaczania zintegrowanego wyniku oraz wyliczania agregatów w oknie o zadanym rozmiarze. Ponadto, zapytania muszą być wykonywane efektywnie, aby nie "gubić" napływających danych. Wiąże się to z kolei z technikami optymalizacji zapytań i zapewnienia właściwego (optymalnego) rozmiaru pamięci RAM. Optymalizacja zapytań na strumieniach danych jest bardzo trudna z badawczego i technicznego punktu widzenia. Sam proces optymalizacji musi być bardzo szybki, a otrzymany plan wykonania zapytania musi być wystarczająco efektywny, aby zapytanie zdążyło przeanalizować napływające dane. Dodatkowo, plan zapytania operującego na wielu strumieniach musi uwzględniać charakterystyki tych strumieni i zależności między nimi. Te wymagania powodują, że proces optymalizacji zapytań na danych strumieniowych jest nadal otwartym problemem badawczym.

Od kilku lat obserwuje się nasilenie badań w zakresie przetwarzania, gromadzenia i analizowania danych napływających w sposób ciągły. Publikacje w tej dziedzinie pojawiają się w wiodących międzynarodowych konferencjach, m.in. SIGMOD, ICDE, VLDB, i czasopismach, m.in. IEEE Computer, IEEE Data Engineering Bulletin, ACM Transactions on Database Systems, VLDB Journal.

W tym kontekście, **tematyka rozprawy doktorskiej dotyczy ważnego i trudnego problemu badawczego. Wpisuje się ona w światowe trendy badawcze i technologiczne w zakresie przetwarzania strumieni danych.**

**Biorąc pod uwagę aktualny stan badań w zakresie przetwarzania strumieni danych, należy stwierdzić, że postawione tezy rozprawy i jej zadania szczegółowe są właściwe dla rozprawy doktorskiej.**

## 2. Struktura rozprawy

Recenzowana rozprawa doktorska składa się z pięciu rozdziałów, bibliografii, jednego dodatku zawierającego opis gramatyki języka analizy strumieni danych oraz spisów symboli, skrótów, ilustracji i tabel. Rozdział 1 zawiera wprowadzenie do problematyki rozprawy, motywację do podjęcia problemu i tezy rozprawy. Rozdział 2 przedstawia stan wiedzy w zakresie zarządzania danymi strumieniowymi. W szczególności, Doktorant skoncentrował się tu na przedstawieniu koncepcji analizy danych strumieniowych, klasyfikacji strumieni i operatorów umożliwiających przetwarzanie danych strumieniowych. Rozdział ten zawiera jednocześnie kontrybucję rozprawy w postaci definicji logicznych operatorów strumieniowych i ich implementacji w postaci konkretnych algorytmów. Rozdział 3 opisuje kolejną kontrybucję rozprawy w postaci języka analizy danych strumieniowych o nazwie StreamAPAS. Rozdział 4 przedstawia opracowaną architekturę prototypowego systemu przetwarzania danych strumieniowych, mechanizm predykcji opóźnień systemu i koncepcję partycjonowania operatorów jako mechanizmu optymalizacji zapytań na danych strumieniowych. Rozdział 5 zawiera podsumowanie rozprawy.

## 3. Ocena rozprawy

### 3.1. Ważność tematyki

Przedmiot recenzowanej rozprawy zalicza się do **ważnego** i **aktualnego** nurtu badań na świecie w dziedzinie zarządzania danymi strumieniowymi. Publikacje w tej dziedzinie ukazują się w materiałach najlepszych konferencji i czasopism międzynarodowych. Podjęta w rozprawie problematyka konstrukcji języka zapytań na strumieniach danych i optymalizacji zapytań stanowią nadal otwarte i ważne problemy badawcze.

### 3.2. Cel rozprawy i jej główne wyniki

Doktorant postawił sobie dwa główne zadania. Po pierwsze, skonstruowanie języka zapytań umożliwiającego analizę danych strumieniowych z możliwością rozszerzania jego funkcjonalności. Po drugie, opracowanie architektury systemu przetwarzania danych strumieniowych zawierającego mechanizmy optymalizacji zapytań.

Do głównych wyników rozprawy realizujących wspomniane wyżej cele zaliczam:

1. formalne definicje operatorów logicznych na danych strumieniowych i ich realizacje w postaci operatorów fizycznych, tj. konkretnych algorytmów implementujących te operatory logiczne;
2. opracowanie języka zapytań StreamAPAS dla danych strumieniowych wraz z implementacją;
3. opracowanie modelu predykcji opóźnień w module przetwarzania zapytań na danych strumieniowych i implementacja symulatora;
4. opracowanie techniki optymalizacji zapytań na strumieniach danych w oparciu o technikę partycjonowania (grupowania) operatorów.

Zdaniem recenzenta oba główne cele rozprawy zostały osiągnięte. Na podkreślenie zasługuje fakt oceny eksperymentalnej wszystkich opracowanych w rozprawie rozwiązań i ich odniesienie do wybranych rozwiązań znanych z literatury światowej.

### 3.3. Uwagi merytoryczne

1. W podejściu do optymalizacji zapytań na danych strumieniowych Doktorant zaproponował optymalizację regułową. Ten rodzaj optymalizacji ma szereg wad (znanych z systemów relacyjnych baz danych) i w praktyce nie jest już stosowany w systemach komercyjnych.

Dlaczego Doktorant zdecydował się na ten rodzaj optymalizacji? Jakich trudności można się spodziewać w podejściu kosztowej optymalizacji zapytań na danych strumieniowych?

W kontekście opracowanego optymalizatora regułowego nasuwają się następujące pytania. Po pierwsze, kiedy jest budowany plan wykonania zapytania - czy jest to tzw. optymalizacja statyczna, w której plan wykonania jest konstruowany przed uruchomieniem zapytania, w procesie jego kompilowania, czy jest to tzw. optymalizacja dynamiczna, w której plan wykonania jest konstruowany na bieżąco w trakcie wykonywania zapytania. W drugim przypadku, plan jest dostosowywany do otrzymywanych wyników pośrednich. Jak wiadomo, oba rozwiązania mają swoje wady i zalety. Po drugie, na ile optymalizacja dynamiczna może być stosowana dla zapytań na strumieniach danych. Ze względu na intensywność strumieni i narzut czasowy optymalizacji, przewiduję tu problemy efektywnościowe.

2. Strona 27, ostatni akapit, punkt b) - na jest realne założenie o identycznym czasie trwania każdego zdarzenia? Strona 28, punkt c) - na jest realne założenie o znanym czasie trwania zdarzenia? W jakich zastosowaniach wspomniane założenia są spełnione?

3. Punkt 2.8 - nie jest jasne, czy tabela historii jest przechowywana na dysku, czy w RAM.

4. Strona 49 i dalsze - zdaniem recenzenta termin "tabela haszująca" został użyty niewłaściwie. Sugeruje on, że tabela dokonuje haszowania. W rzeczywistości haszowanie jest realizowane za pomocą funkcji, a jego wyniki być może znajdują się w tabeli. Jak rozumiem, w algorytmie 2.2 $H_{PK}$ jest adresem w pamięci.

5. Strona 67 - dlaczego połączenie (ang. join) implementuje się tu jako operację selekcji na iloczynie kartezjańskim? Taka implementacja jest niezwykle nieefektywna. Na tej samej stronie, w punkcie 1) listy wypunktowanej została omówiona reguła przeniesienia operatora iloczynu kartezjańskiego na początek planu wykonania zapytania. Takie podejście odbiega od uznanego w praktyce podejścia minimalizowania pośredniego wyniku zapytania jak najwcześniej to jest możliwe.

6. W praktyce w systemach baz (hurtowni) danych czasu rzeczywistego wykorzystuje się dwie główne miary jakości systemu, tj. Quality of Data (QoD) i Quality of Service (QoS). Coraz częściej proponuje się też miarę zagregowaną odwzorowującą oczekiwania użytkownika, tzw. Quality of Expectation (QoE). Dlaczego w opisie wymagań dla zaproponowanego języka zapytań pominięto QoD?

7. Punkt 3.15 wprowadza strukturę R-drzewa definiowaną na strumieniu danych. Przy założeniu, że strumień danych napływa w sposób ciągły, jaki jest sens definiowania jakiegokolwiek indeksu na strumieniu? Indeks taki będzie musiał być na bieżąco uaktualniany. Czy nie zachodzi obawa utraty mocy obliczeniowej na uaktualnianie indeksu i czy system zdąży uaktualnić indeks?

8. W rozprawie zaproponowano język zapytań i architekturę prototypowego systemu przetwarzania danych strumieniowych. W pracach naukowych z zakresu real-time data warehouse/ near real-time data warehouse / right-time data warehouse podkreśla się potrzebę wykonywania zapytań na strumieniach danych i w pewnych przypadkach jednocześnie na tradycyjnej hurtowni danych. Czy taka funkcjonalność jest wspierana w rozwiązaniu zaproponowanym w recenzowanej rozprawie doktorskiej? Zdaniem recenzenta funkcjonalność ta nie jest wspierana.

9. Po przeczytaniu rozprawy niejasny pozostaje sposób wykonywania n równoczesnych zapytań na tym samym strumieniu. Czy strumień jest rozszczepiany na n identycznych strumieni, po jednym dla każdego zapytania, czy też zapytania są kolejkowane z uwzględnieniem kryterium QoS, czy też jest stosowany mechanizm pipe-lining?

10. Punkt 4.1.8 - nie jest jasne na jakiej podstawie przyjęto reguły opisane na końcu strony 150.

11. Jak wyglądałby wzór (4.4) dla n strumieni, każdy z niezależnym średnim czasem odstępu grup krotek w strumieniu?

12. W punkcie 4.3.3 zaproponowano algorytmy podziału operatorów na partycje. Nie jest jasne kiedy i jak często te algorytmy są uruchamiane. Czy dostosowują one podział na partycje do aktualnego obciążenia? Jaka jest złożoność obliczeniowa tych algorytmów. Dlaczego nie podano ich pseudokodu?

## 3.4. Uwagi edytorskie

Pod względem edytorskim, rozprawa zawiera drobne błędy. Szczegółowy ich wykaz został zamieszczony poniżej.

1. W rozdziale 2 znajduje się opis stanu wiedzy w zakresie tematyki rozprawy i częściowa kontrybucja rozprawy w postaci definicji operatorów. Taki układ, utrudnia stwierdzenie co jest opisem stanu wiedzy, a co juz kontrybucją rozprawy.
2. W rozdziale 5 jest jeden podrozdział 5.1 - w takim przypadku nie wyróżniamy podrozdziałów. Czynimy to, gdy jest ich przynajmniej 2.
3. Strona 8, akapit 2 i strona 9 akapit 1: niewłaściwie zastosowano pojęcie strumieniowa baza danych SDMS. Baza danych to nie to samo co system zarządzania bazą danych.
4. Strona 11, ostatni akapit - w rozprawie nie ma rozdziału 6.
5. Strona 12, akapit 1 - czy "tabela rekordów" jest tym samym co relacja?
6. Punkt 2.7.1, pod wzorem 2.9 - "Operatora" zamienić na "Operator".
7. W celu zachowania symetrii podpunktów punktu 2.7 sugerowałbym podanie algorytmów dla wszystkich omawianych tu operatorów.
8. Strona 41, drugi element listy punktowanej - usunąć spację po $t.t_s$.
9. Strona 44 - wprowadzono tu pojęcia "tabela historii", "tablica historii", "struktura lokalna H", "kolekcja H", a na stronie 47 - "kolekcja krotek H" oznaczające, jak rozumiem to samo. Sugeruję stosowanie jednego pojęcia.
10. Strona 45, akapit 1 - "z struktury" zamienić na "ze struktury".
11. Strona 52, linia 2 - "całkowita większa" zamienić na "całkowitą większą".
12. Strona 80, linia 2 - "Hypernion" zamienić na "Hyperion".
13. Strona 95, akapit 4 - "Obiek" zamienić na "Obiekt".
14. Punkt 3.7, linia 2 - "zdefiniowanie" zamienić na "zdefiniowania".
15. Strona 122, ostatni akapit - elementy listy numerowanej nie pasują stylistycznie do początku zdania "Do brakujących elementów definicji zalicza się:".
16. Strona 124, akapit 2, linia 3 - zamienić ";" na "," w "skladowych;".
17. W całej pracy: często brakuje spacji przed referencją bibliograficzną [], np. na tronie 131.
18. Strona 134 - uspójnić pisownię "mikro jądro", "mikrojądro".
19. Strona 139 - dwa ostatnie zdania są stylistycznie niepoprawne.
20. Strona 150, akapit 2 - zdanie 4 i 5 są stylistycznie niepoprawne.
21. Strona 151, linia 3 od dołu - zamienić "łącznia" na "łączenia".
22. Strona 155, akapit 2, linia 3 - zdanie "Aby wydajność ..." jest niepoprawne stylistycznie.
23. Strona 158, linia 8 od góry - zdanie "Średni czas ..." jest niepoprawne stylistycznie.
24. Strona 168, ostatnia linia - zamienić "informuej" na "informuje".

## 4. Ocena końcowa i rekomendacja

Podsumowując recenzję, uważam, że cel rozprawy mgra Aleksandra Chrószcza został osiągnięty. Po pierwsze, Doktorant opracował koncepcję języka zapytań na strumieniach danych wraz z formalną definicją operatorów i ich implementacją. Po drugie, zaproponował mechanizm optymalizacji zapytań na strumieniach danych w postaci techniki grupowania operatorów. Po trzecie, zaimplementował prototypowy system i dokonał eksperymentalnej oceny zaproponowanych rozwiązań w odniesieniu do wybranych rozwiązań konkurencyjnych. Opracowane definicje i mechanizmy są podparte zaawansowanym aparatem matematycznym.

Problematyka przetwarzania danych strumieniowych podjęta w rozprawie jest trudna ze względu na silne ograniczenia czasowe w module wykonywania zapytań a także ze względu na liczność strumieni danych, które należy analizować. Osiągnięte w recenzowanej rozprawie wyniki są dobrym punktem wyjścia do rozbudowania koncepcji i prototypowego systemu.

Pomimo wymienionych wyżej uwag krytycznych, które mają charakter dyskusyjny, **uważam, że recenzowana rozprawa doktorska mgra Aleksandra Chrószcza spełnia wymagania stawiane rozprawom doktorskim przez obowiązującą ustawę, wobec czego wnoszę o dopuszczenie jej do publicznej obrony.**

Dorobek publikacyjny Doktoranta objęty zakresem rozprawy spełnia wymagania Rady Wydziału Automatyki, Elektroniki i Informatyki Politechniki Śląskiej odnośnie do wyróżnienia rozprawy. W związku z tym, wnioskuję także o wyróżnienie niniejszej rozprawy doktorskiej.

dr hab. inż. Robert Wrembel, prof. nadzw.