Warszawska Wyższa
Szkoła Informatyki

Warszawa 2011

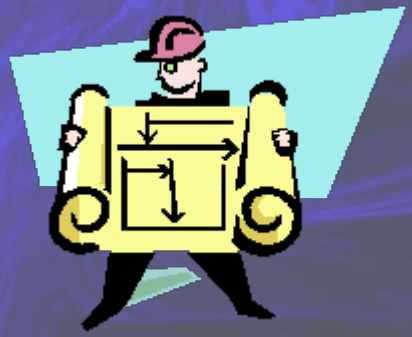
Optymalizacja wydajności zapytań w testowaniu schematu bazy danych

Autor: Hubert Kwiatkowski

Promotor: dr inż. Paweł Figat, mgr inż. Andrzej Ptasznik

Plan prezentacji

- Wstęp - dziedzina przedmiotowa
- Cel pracy
- Część teoretyczna
  - Relacyjne bazy danych
  - Technologie: XML, wyszukiwanie pełnotekstowe, HIERARCHYID
- Część praktyczna
  - Opis problemu
  - Schematy bazy danych
  - Typowe zapytania
  - Optymalizacja zapytań
  - Testy wydajności
  - BD Tester
- Podsumowanie

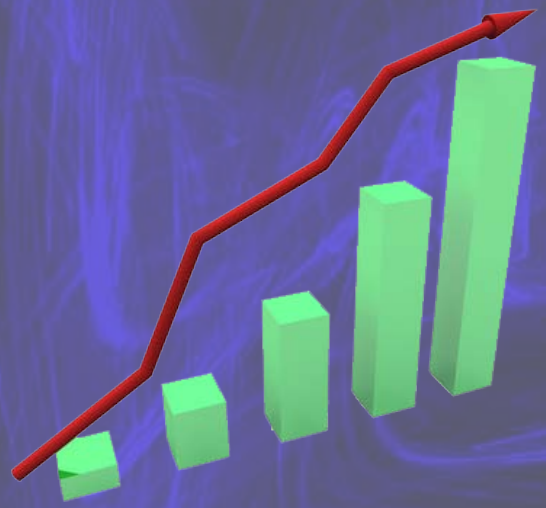


Wstęp: optymalizacja zapytań

- Cel: zwiększenie wydajności podsystemu bazodanowego
- Kryteria wydajności: czas realizacji zapytań, przepustowość, skalowalność

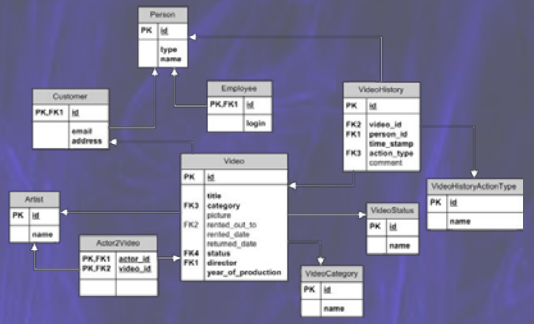
Proces optymalizacji:

- Projekt struktury bazy danych
- Dostrajanie zapytań
- Dobór indeksów
- Poziom współbieżności – blokady
- Wąskie gardła sprzętowe – procesor, pamięć operacyjna i masowa



Cel pracy

Celem pracy jest analiza wydajności zapytań dla różnych wariantów projektowych schematu bazy danych.



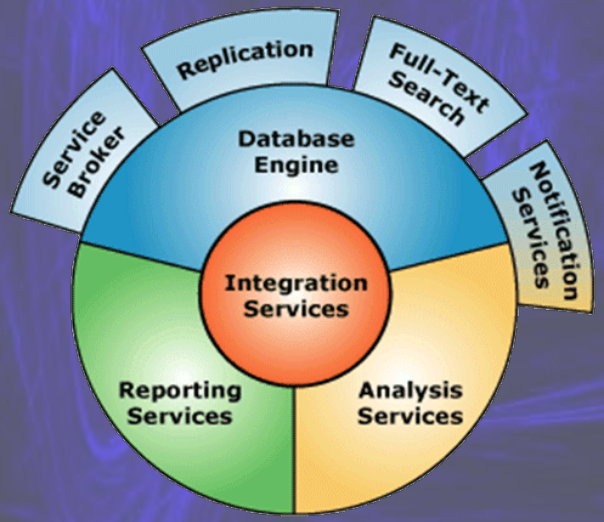
Odpowiedź na pytanie:

Czy klasyczny schemat bazy danych daje najlepsze rezultaty pod względem wydajności zapytań?

Relacyjne bazy danych

- Rys historyczny - model relacyjny Codda, język SQL
- Normalizacja danych
- Logiczna i fizyczna organizacja danych
- Proces realizacji zapytania
- Optymalizacja zapytań

Microsoft® SQL Server® 2008



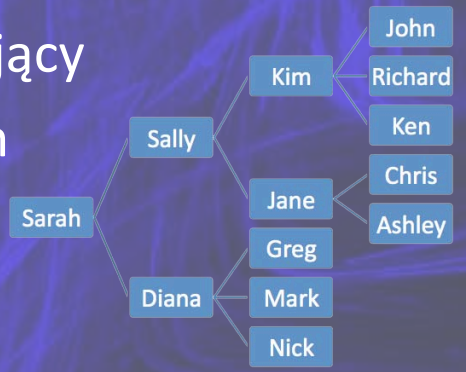
Technologie

XML – uniwersalny język formalny służący do reprezentowania różnych danych w strukturalizowany sposób *[Wikipedia]*

<?xml?>

Wyszukiwanie pełnotekstowe – usługa umożliwiająca przeszukiwanie dużych fragmentów tekstów pod kątem znaczeń poszczególnych słów i całych fraz.

HIERARCHYID – obiektowy typ danych, umożliwiający przechowywanie i obsługę danych hierarchicznych



Opis problemu

- Wielodziedzinowy system ogłoszeniowy (odpowiednik funkcjonalny Otomoto)
- Przedmiot ogłoszenia scharakteryzowany dziesiątkami cech
- Hierarchiczna struktura kategorii ogłoszeń

- Problemy:
  - Efektywne wyszukiwane ogłoszeń po ich cechach, przykładowo: 100 tys. ogłoszeń * 30 cech = 3 mln wierszy
  - Trudności w oszacowaniu docelowego obciążenia systemu (wraz ze wzrostem popularności serwisu liczba internautów będzie rosła)

Schematy baz danych

Eksperymentalny dobór możliwych do zastosowania wariantów schematów baz danych:

- Wariant klasyczny (3 PN, 2 warianty zapytań)
- Wariant zdenormalizowany
- Wariant XML
  - Formatowanie nie uwzględniające typów danych
  - Formatowanie uwzględniające typy danych (walidacja XML Schema)
- Wariant z wyszukiwaniem pełnotekstowym
  - Cechy liczbowe w kolumnach
  - Identyfikacja możliwych przedziałów, przypisanie unikalnych identyf.

Typowe zapytania

Dwie typowe grupy zapytań obciążających bazę danych:

- Zapytania o „długim” czasie realizacji
- Zapytania o „krótkim” czasie realizacji występujące często

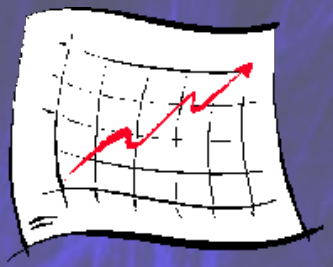
Typowe zapytania dla przedstawionego problemu:

- Wyszukiwanie ogłoszeń spełniających ustalone kryteria (cechy, kategorie, inne atrybuty ogłoszenia np. cena)
- Spis ogłoszeń w danej kategorii podzielony na podstrony (podczas przeglądania ogłoszeń na stronie www)
- Pobieranie wszystkich informacji o danym ogłoszeniu

Optymalizacja zapytań

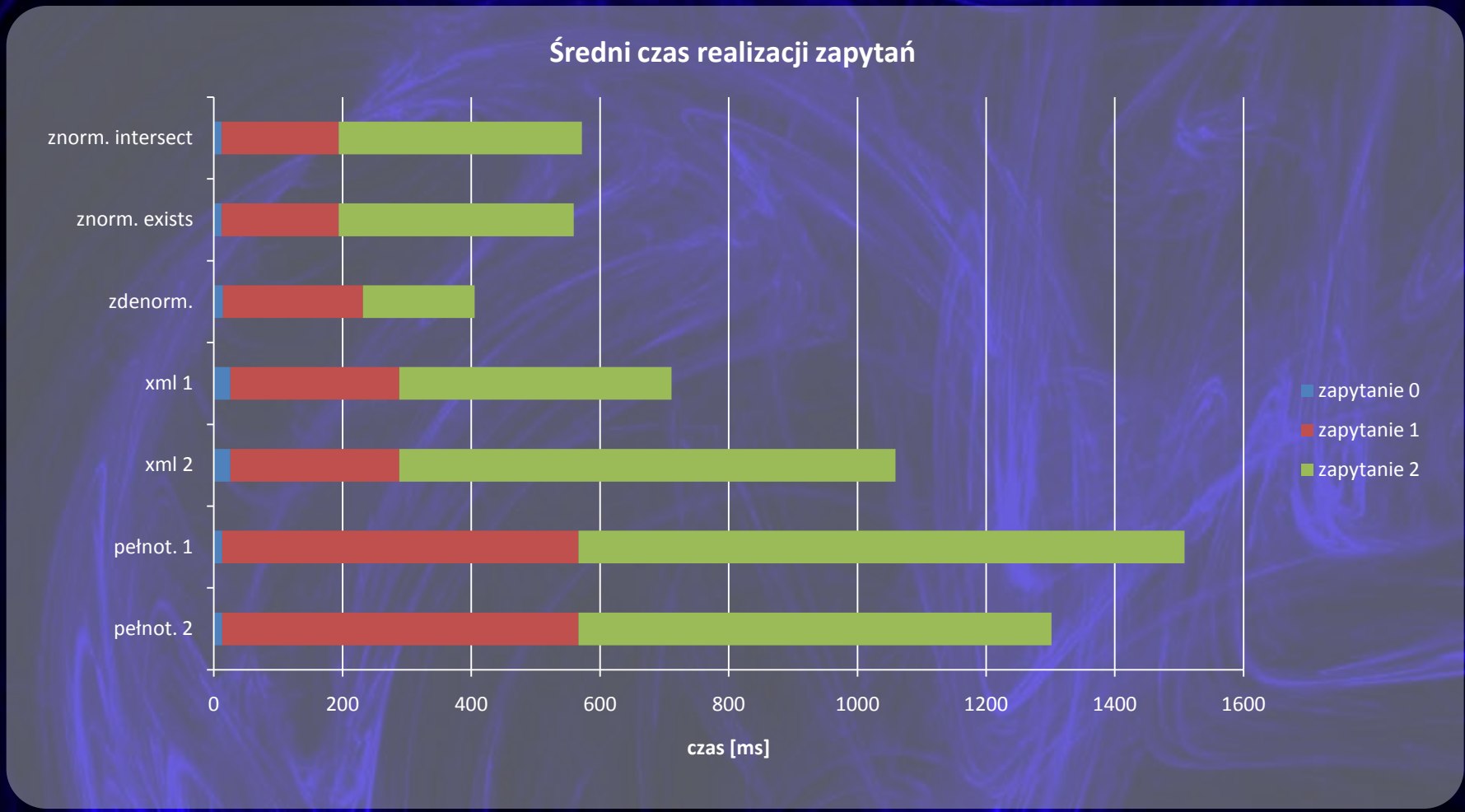
- Cel: wyznaczenie jak najbardziej efektywnego planu realizacji zapytania (kryterium – czas realizacji zapytania)

Realizacja:



- Przygotowanie wstępnych wersji 15 zapytań
- Analiza planu wykonania zapytania – wyszukanie „wąskich gardeł” wydajnościowych
- Dostrajanie zapytania – przeredagowanie fragmentów zapytania sprawiających trudności optymalizatorowi
- Dobór i dostrajanie indeksów

Wyniki - optymalizacja



Testy wydajności

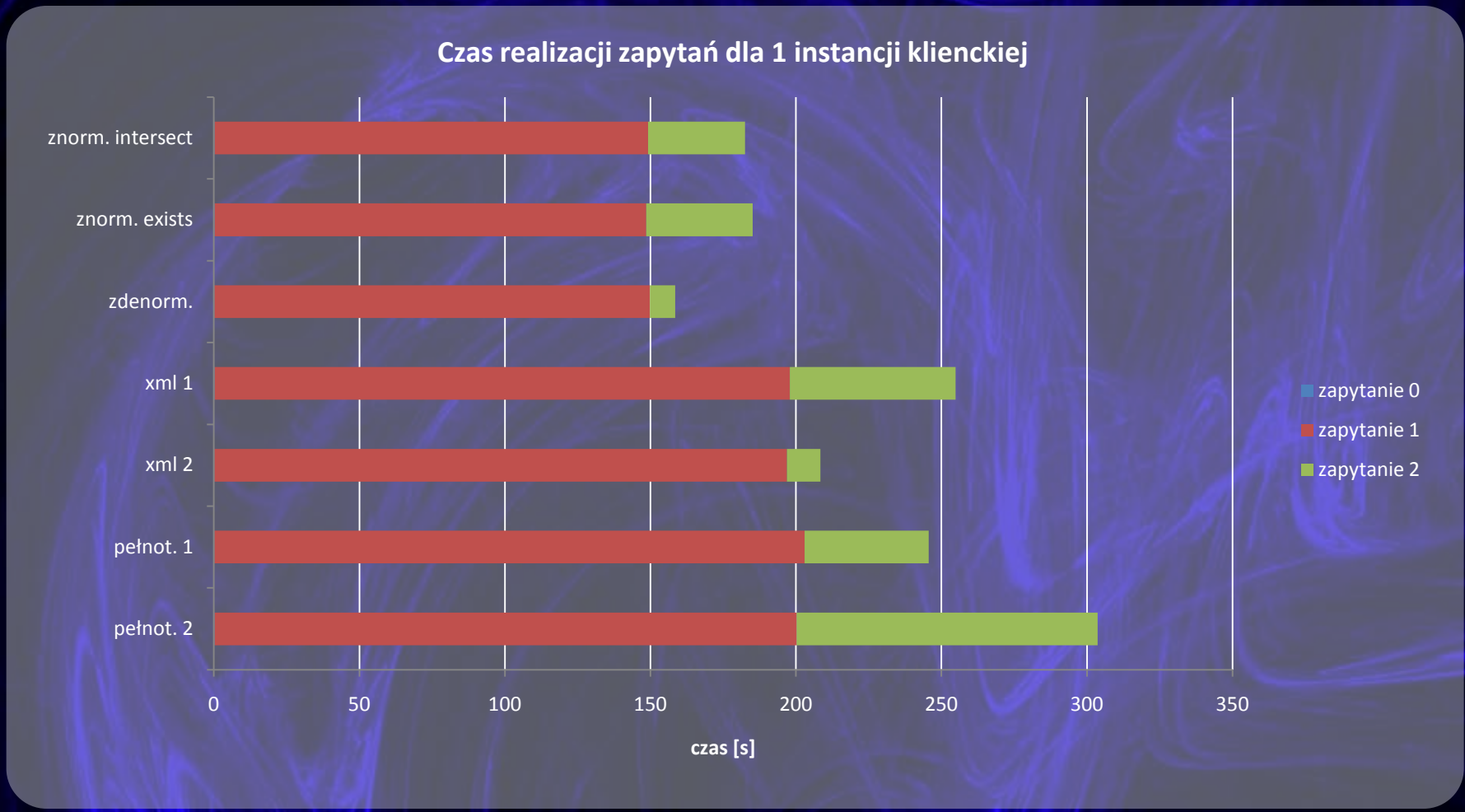
- Przygotowanie danych testowych
  - Wykorzystanie danych zbliżonych do rzeczywistych
- Procedura testowa
  - Minimalizacja wpływu czynników zewnętrznych
  - Uśredniony wynik z kilku pomiarów
  - Czyszczenie pamięci podręcznej (dane, plany zapytań)
- Pomiar wydajności zapytań
  - Zoptymalizowane zapytania dla poszczególnych wariantów schematów baz danych
  - Pomiar dla najlepszego i najgorszego przypadku (cache)
  - Skalowalność każdego rozwiązania – testy wieloużytkownikowe

BD Tester

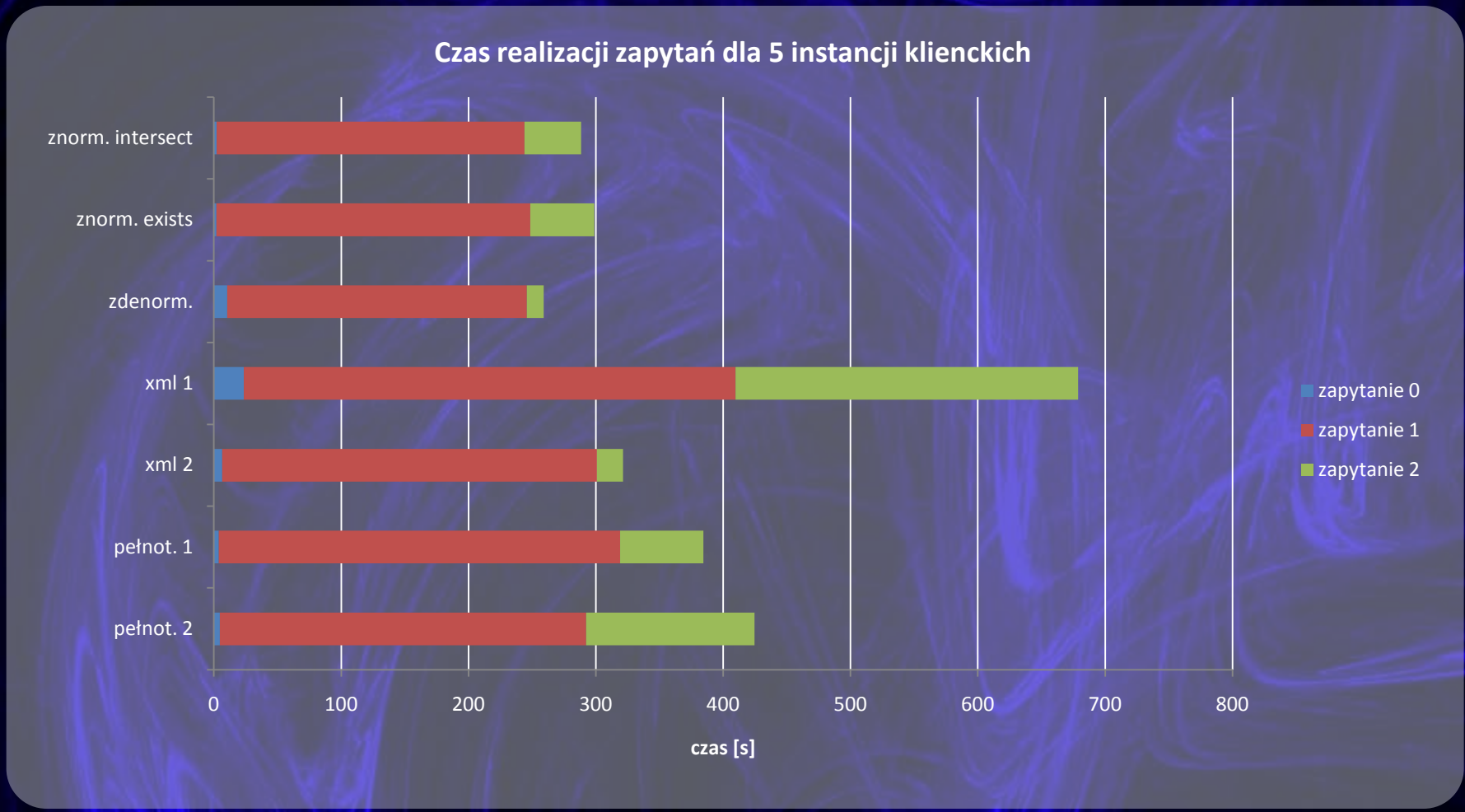
- Aplikacja umożliwiająca przeprowadzenie wieloużytkownikowych testów obciążeniowych bazy danych
- Realizacja – .NET C# 3.5, technologia klient-serwer

- Możliwości aplikacji:
  - Generowanie losowych sekwencji testowych wg zdefiniowanego scenariusza testowego
  - Konfigurowanie przebiegu testu tylko na maszynie serwera
  - Synchronizacja przebiegu testów pomiędzy wieloma maszynami klienckimi
  - Komunikacja z maszynami klienckimi w celu odebrania wyników testów

Wyniki – testy obciążeniowe



Wyniki – testy obciążeniowe



Podsumowanie

Na podstawie przeprowadzonego procesu optymalizacji zapytań oraz testów obciążeniowych można stwierdzić, że najwydajniejszy jest zdenormalizowany wariant schematu bazy danych, a tym samym postawiona na wstępie hipoteza jest błędna. Jednakże posiada on wiele wad funkcjonalnych, które ograniczają jego praktyczne zastosowanie:

- Anomalie aktualizacji
- Utrata części informacji (np. cechy typu pole wyboru)
- Konieczność użycia instrukcji DDL zamiast DML (np. dodanie cechy)
- Nadzór nad indeksami i ich czasochłonne przebudowy

Dziękuję za uwagę.