BO·TT·EGA
IT minds

Program szkolenia:

Bazy danych w projektowaniu wysoko wydajnych systemów rozproszonych

Informacje:

Nazwa:	**Bazy danych w projektowaniu wysoko wydajnych systemów rozproszonych**
Kod:	**ddb-database**
Kategoria:	Bazy danych
Odbiorcy:	developerzy, architekci
Czas trwania:	3 dni
Forma:	50% wykłady / 50% warsztaty

Poziom skomplikowania i mnogość dostępnych rozwiązań zwiększyła się w sposób wykładniczy, a ryzyko wyboru niewłaściwej bazy danych może okazać się katastrofą całego projektu.

Mikrousługi nad którymi pracowaliśmy x czasu zostały wdrożone. Skrupulatnie wydzieliliśmy konteksty, opanowaliśmy granice agregatów, event storming sprawił, że naprawdę wiemy o co w tym biznesie chodzi. Jednak po wdrożeniu na produkcję system okazuje się niewydajny, może trzeba zmienić bazę danych? A dlaczego w zasadzie zdecydowaliśmy się na bazę relacyjną? Może dokumenty będą lepsze? No właśnie, w nowoczesnych, rozproszonych systemach nie jest to takie oczywiste..

Dlatego to szkolenie jest dla Ciebie jeżeli chcesz:

- poznać drivery architektoniczne wskazujące na wybór odpowiedniej bazy danych do projektowanego systemu
- wiedzieć jak baza danych z której korzystasz działa “pod maską”
- poznać wzorce aplikacyjne pozwalające opanować błędy związane z brakiem spójności danych

Szkolenie to niezbędnik każdego specjalisty projektującego, rozwijającego i utrzymującego aplikacje zbudowane w architekturze mikrousługowej.

Uczestnicy zostaną wprowadzeni do podstawowych pojęć związanych z systemami rozproszonymi i bazami danych. Dowiedzą się, dlaczego projektowanie wysoko wydajnych systemów rozproszonych jest kluczowe w dzisiejszych aplikacjach. Warsztat skupia się na praktycznym projektowaniu wykorzystania baz danych w kontekście systemów rozproszonych. Uczestnicy dowiedzą się, jak zoptymalizować wydajność swoich baz danych i unikać typowych pułapek. Szkolenie będzie zawierać wykłady teoretyczne, studia przypadków, praktyczne ćwiczenia oraz dyskusje, które pomogą uczestnikom zrozumieć i zastosować teorię w praktyce. Po zakończeniu szkolenia uczestnicy będą mieć solidne podstawy w projektowaniu baz danych w systemach rozproszonych, co pozwoli im efektywniej tworzyć wysoko wydajne aplikacje i systemy.

BOTTEGA IT minds

Szczegółowy program:

1. Wprowadzenie do systemów rozproszonych

- 1.1. Czym są systemy rozproszone?
- 1.2. Bazy danych jako systemy rozproszone
- 1.3. Dlaczego systemy rozproszone są ważne?
- 1.4. Wyzwania w budowaniu systemów rozproszonych

2. Bazy danych relacyjne i nierelacyjne

- 2.1. Relacyjne bazy danych
  - 2.1.1. Architektura relacyjnych baz danych
    - 2.1.1.1. Transaction Log
    - 2.1.1.2. Indeksy
  - 2.1.2. Skalowanie relacyjnych baz danych
    - 2.1.2.1. Skalowanie wertykalne
    - 2.1.2.2. Repliki do odczytu
    - 2.1.2.3. Partycjonowanie
  - 2.1.3. Kiedy stosować relacyjne bazy danych?
  - 2.1.4. Kluczowe czynniki architektoniczne decydujące o wyborze relacyjnej bazy danych
  - 2.1.5. Przykłady systemów opartych o relacyjne bazy danych
- 2.2. Bazy danych NoSQL
  - 2.2.1. Geneza powstania nierelacyjnych baz danych
  - 2.2.2. Rodzaje nierelacyjnych baz danych
    - 2.2.2.1. Key-value
    - 2.2.2.2. Dokumentowe
    - 2.2.2.3. Kolumnowe

BO·TT·EGA IT minds

2.2.2.4. Grafowe

2.2.2.5. Timescale

2.2.2.6. Search Engine

2.2.3. Architektura nie relacyjnych baz danych

2.2.3.1. Twierdzenie CAP

2.2.3.2. Typowe wzorce wykorzystywane w projektowaniu rozproszonych baz danych

2.2.3.2.1. Partitioning

2.2.3.2.2. Sharding

2.2.3.2.3. Replikacja

2.2.3.2.4. Secondary indexes

2.2.3.2.5. Lider, follower

2.2.4. Kluczowe czynniki architektoniczne decydujące o wyborze nierelacyjnych bazy danych

2.2.5. Przykłady systemów opartych o nierelacyjne bazy danych

2.3. Zarządzanie spójnością danych w nierelacyjnych bazach danych

2.3.1. Gwarancja spójności Eventual Consistency

2.3.2. Gwarancja spójności Strong Consistency

2.4. Zrozumienie i łagodzenie typowych problemów związanych z rozproszeniem danych

2.4.1. Read-Write Quorums

2.4.2. Read Your Own Writes

2.5. Strategie odzyskiwania danych po awarii i tworzenia kopii zapasowych

2.5.1. Ostatni wygrywa

2.5.2. Wektor wersji

2.5.3. Conflict Free Replicated Data Types (CRDT)

3. System rozdystrybuowanego logu

BO·TT·EGA
IT minds

- 3.1. Wzorzec pub/sub
- 3.2. Apache Kafka jako baza danych
  - 3.2.1. Dualizm strumień-tablica
  - 3.2.2. Kompaktowanie danych
- 3.3. Procesowanie danych w czasie rzeczywistym vs klasyczne hurtownie danych
- 3.4. Przykłady systemów opartych distributed log
  - 3.4.1. aplikacji chat messaging system

4. Projektowanie wysoko wydajnych aplikacji wykorzystujących nierelacyjne bazy danych

- 4.1. Najczęstsze problemy
  - 4.1.1. Transakcje rozproszona
  - 4.1.2. Zapytania skanujące pełny zbiór danych
- 4.2. Wzorce wykorzystywane w projektowanie aplikacji odpornych na błędy
  - 4.2.1. CQRS
  - 4.2.2. Inbox, Outbox
  - 4.2.3. Two-Phase Commit (2PC)
  - 4.2.4. Saga
  - 4.2.5. Change Data Capture (CDC)
  - 4.2.6. Circuit Breaker
  - 4.2.7. Read Repair

5. Omówienie implementacji wybranych rozproszonych baz danych

- 5.1. Bazy danych klucz wartość
  - 5.1.1. Redis
  - 5.1.2. DynamoDB
  - 5.1.3. Voldemort
- 5.2. Bazy danych dokumentowe

BO·TT·EGA
IT minds