BOTTEGA IT minds

Program szkolenia:

Zaawansowana architektura systemów PHP - projektowanie i implementacja skalowalnych aplikacji webowych

Informacje:

Nazwa:	**Zaawansowana architektura systemów PHP - projektowanie i implementacja skalowalnych aplikacji webowych**
Kod:	**PHP-arch**
Kategoria:	PHP
Odbiorcy:	developerzy
Czas trwania:	3 dni
Forma:	70% wykłady / 30% warsztaty

Szkolenie zostało przygotowane z myślą o programistach i architektach pracujących nad dużymi aplikacjami webowymi typu mission-critical, funkcjonujących w oparciu o klaster redundantnych serwerów.

Szkolenie zawiera szereg praktycznych rozwiązań opracowanych na podstawie doświadczeń w dużych projektach webowych, hostowanych na klastrach wielkości 10-600 fizycznych serwerów.

Zalety szkolenia:

- Zaingrowane podejśćie do: skalowania, bezpieczeństwa, failover, CQRS
- Całościowe i szersze spojrzenie na problemy
- Dostęp do wiedzy eksperckiej

BOTTEGA IT minds

Szczegółowy program:

1. Wprowadzenie

- 1.1. Dlaczego warto zawczasu zadbać o skalowanie aplikacji?
- 1.2. Historie popularnych witryn internetowych, ich dobre i złe wybory
  - 1.2.1. Zestawienie najczęściej występujących problemów
- 1.3. Korzyści wynikające z zastosowania architektury umożliwiającej skalowanie
- 1.4. Skalowalność a wydajność systemu

2. Architektura kodu aplikacji

- 2.1. Backend i frontend aplikacji
  - 2.1.1. Separowalność i specjalizacja poszczególnych części aplikacji
  - 2.1.2. Komunikacja
- 2.2. Organizacja kodu
  - 2.2.1. Architektura warstwowa oraz Ports and Adapters
  - 2.2.2. Moduły oprogramowania
    - 2.2.2.1. Możliwości i zastosowanie Composera
  - 2.2.3. Wzorce projektowe
  - 2.2.4. Serwisy usługowe i Dependency Injection
  - 2.2.5. Mapowanie relacyjno-obiektowe
    - 2.2.5.1. Przegląd dostępnych rozwiązań
    - 2.2.5.2. Ograniczenia narzędzi ORM
  - 2.2.6. Command-query Responsibility Segregation
    - 2.2.6.1. Stos Command
    - 2.2.6.2. API dla modelu biznesowego
    - 2.2.6.3. Stos Read

2.2.6.4. Optymalizacja pod kątem dużej ilości odczytów

2.2.6.5. Synchronizacja stosów

2.2.6.6. SQL, widoki, zdarzenia biznesowe

2.2.7. Struktura monolityczna vs architektura mikroserwisów

2.2.7.1. Case studies projektów migrujących pomiędzy stylami, przyczyny i osiągane rezultaty

3. Architektura systemu

3.1. Środowisko uruchomieniowe aplikacji, kod źródłowy to nie wszystko

3.2. Typowa ścieżka skalowania aplikacji

3.2.1. Środowisko single-server jako punkt startowy

3.2.2. Rozproszenie usług w pojedynczych instancjach

3.2.3. Rozproszenie usług w wielu instancjach

3.2.4. Przegląd typowych problemów

3.3. Redundancja usług dla wysokiej dostępności

3.3.1. Kierowanie ruchem pomiędzy reduntantnymi serwerami

3.3.2. Obsługa sytuacji krytycznych, failover

3.3.3. Wpływ redundancji na kod aplikacji

3.4. Eliminacja obciążeń

3.4.1. Przyśpieszanie aplikacji PHP

3.4.1.1. Opcode cache

3.4.1.2. Buildy aplikacji

3.4.2. Cache'owanie danych

3.4.2.1. Wprowadzenie do cache'owania

3.4.2.2. Tworzenie i inwalidacja cache

3.4.2.3. Redis

3.4.2.4. Techniki wykorzystania dedykowanych narzędzi w celu odciążenia warstwy bazy danych

3.4.3. Cache HTTP

3.4.3.1. Wprowadzenie do cache'owania

3.4.3.2. Varnish jako web-application accelerator

3.5. Skalowanie warstwy bazy danych

3.5.1. Modele replikacji

3.5.1.1. Master-slave(s)

3.5.1.2. Master-master

3.5.1.3. Problemy aplikacyjne wynikające z replikacji

3.5.1.4. Zakleszczenia

3.5.1.5. Replication-lag

3.5.2. Shardowanie tabel

3.5.2.1. Wprowadzenie do koncepcji shardowania

3.5.2.2. Korzyści i ograniczenia

3.5.2.3. Techniki podziału baz danych

3.5.2.4. Konsekwencje dla warstwy dostępu do danych w aplikacji

3.5.3. Rozwiązania NoSQL

3.5.3.1. CAP theorem

3.5.3.2. Przegląd dostępnych rozwiązań i ich możliwości

3.6. Systemy kolejkowe

3.6.1. Wprowadzenie do systemów kolejkowych

3.6.2. Wykorzystanie systemów kolejkowych do rozdzielania systemów

3.6.3. Narzędzia

3.6.3.1. AMQP

3.6.3.2. RabbitMQ

3.7. Deployment aplikacji na klaster serwerowy

3.7.1. Różnice pomiędzy deploymentem w ramach pojedynczego serwera a klastrem

3.7.2. Konsekwencje deploymentu systemów wielowarstwowych

3.8. Monitoring

3.8.1. Monitoring pracy infrastruktury serwerowej

3.8.2. Monitoring pracy aplikacji

3.8.3. Narzędzia

3.9. Infrastruktura serwerowa

3.9.1. Cloud computing vs infrastruktura dedykowana

4. Case-studies realnych projektów

4.1. Shardowane bazy danych

4.2. Skalowalne systemy kolejkowe

4.3. Skalowanie aplikacji w sytuacjach wzmożonego ruchu