BO·TT·EGA
IT minds

Program szkolenia:

Hiperwydajne systemy rozproszone

Informacje:

Nazwa:	**Hiperwydajne systemy rozproszone**
Kod:	**Arch-hiper**
Kategoria:	Architektura systemów i aplikacji
Czas trwania:	3 dni
Forma:	25% wykłady / 75% warsztaty

Praktyczne warsztaty z tworzenia wydajnych i skalowalnych systemów rozproszonych z wykorzystaniem bibliotek Akki.

Tworzenie skalowalnych systemów rozproszonych nie jest proste, ale istnieją proste abstrakcje i wzorce, które mogą w tym pomóc. Korzystając z stosu technologicznego Akki możemy tworzyć systemy, które gotowe są na gwałtowny wzrost ruchu i są w stanie efektywnie (koszty, czasy odpowiedzi, przepustowość) obsłużyć 1 000, 10 000 oraz 1 000 000 req/s.

Akka nie jest frameworkiem, jest to zbiór bibliotek, które można dowolnie łączyć i konfigurować oraz używać z innymi rozwiązaniami, takimi jak np. Spring.

Szkolenie koncentruje się na poznawaniu wielu aspektów budowania wydajnych i skalowalnych rozwiązań, poprzez tworzenie przykładowej aplikacji, która ma być: zawsze dostępna, odporna na błędy, elastyczna na zmiany i zorientowana na zdarzenia. Innymi słowy ma podążać za zasadami zebranymi w Reactive Manifesto.

Wzorce takie jak Event Sourcing i CQRS będą naszymi podstawowymi narzędziami, które dogłębnie przeanalizujemy i wykorzystamy za pomocą gotowych i wygodnych komponentów.

Omówimy jak można skalować każdą cześć systemu całkowicie niezależnie i dynamicznie (w zależności od obsługiwanego ruchu). Baza danych (nawet relacyjna), w końcu przestanie być wąskim gardłem. Duży nacisk położony zostanie na obsługę sytuacji awaryjnych i izolowanie ich wpływu na działanie całości systemu.

Szkolenie możliwe jest do przeprowadzenia w Javie lub Scali.

Zalety szkolenia:

- Proste abstrakcje rozwiązują skomplikowane problemy.
- Zmiana mindsetu w modelowaniu domeny - większy nacisk na modelowanie eventów niż agregatów.
- Event Sourcing może być łatwym i zabójczo skutecznym narzędziem w podejściu event-driven, szczególnie w połączeniu z CQRS.
- Domena jest odseparowana od Akki oraz warstwy infrastrukturalnej przez co jest łatwa w utrzymaniu, testowaniu oraz portowalna.

BO·TT·EGA
IT minds

Szczegółowy program:

1. Dlaczego reaktywność

- 1.1. reactive manifesto

2. Reaktywne mikroserwisy

- 2.1. izolacja stanu, obszaru, czasu i błędów
- 2.2. reaktywne wzorce wykorzystywane w mikroserwisach

3. Korzystanie z Akki w Springu

- 3.1. koegzystencja rozwiązań

4. Model aktorów

- 4.1. stos bibliotek Akki
- 4.2. podstawy aktorów
- 4.3. Actor System
- 4.4. testowanie aktorów
- 4.5. cykl życia aktorów
- 4.6. obsługa błędów w aktorach

5. Event Sourcing

- 5.1. wady i zalety Event Sourcingu
- 5.2. CQRS (zapis danych)
- 5.3. Event Driven vs Event Sourcing
- 5.4. różne sposoby implementacji Eventy Sourcingu
  - 5.4.1. spójność danych: optimitic locking vs single writer principle
- 5.5. Command Sourcing

6. Akka Persistence Typed

- 6.1. podstawowe koncepcje

BOTTEGA IT minds

- 6.2. architektura
- 6.3. separacja domeny od warstwy aplikacyjnej

7. Implementacja agregatu domenowego

- 7.1. podstawowy przepływ domenowy: komenda -> agregat -> zdarzenia
- 7.2. podstawowy przepływ aplikacyjny: serwis -> aktor/encja persystentna -> agregat -> baza danych
- 7.3. stan aktora persystentnego
- 7.4. API domeny
- 7.5. API encji persystentnej

8. Gwarancje dostarczania wiadomości

- 8.1. at-most-once
- 8.2. at-least-once
- 8.3. exactly-once

9. Cykl życia agregatu w Event Sourcingu

- 9.1. wczytywanie stanu
  - 9.1.1. snapshotting
- 9.2. odpytywanie o stan

10. Modelowanie domeny w Event Sourcingu

- 10.1. podejście event-first
- 10.2. wzbogacanie zdarzeń

11. Akka Projections

- 11.1. CQRS (odczyt danych)
- 11.2. konfiguracja oraz sposoby uruchamiania
- 11.3. reaktywne projekcje (reactive streams)

12. Transakcje między wieloma agregatami - wzorzec Sagi

- 12.1. choreografia

BOTTEGA IT minds

- 12.2. orkiestracja

13. Serializacja danych

- 13.1. omówienie rozwiązań do serializacji danych
  - 13.1.1. schema first vs code first
- 13.2. wersjonowanie i ewolucja modelu zdarzenia
  - 13.2.1. kompatybilność wstecz i wprzód

14. Skalowanie bazy danych SQL

15. Skalowanie projekcji

16. Pluginowość warstwy persystencji

- 16.1. dziennik zdarzeń w SQL i NoSQL (Cassandra, DynamoDB)

17. Skalowanie systemów rozproszonych

- 17.1. spójność vs dostępność
- 17.2. wydajność vs skalowalność
- 17.3. contention i prawo Amdahla
- 17.4. koherencja i prawo Gunthera
- 17.5. przepustowość a czasy odpowiedzi

18. Stateful vs stateless systems

19. Akka Cluster

- 19.1. podstawowa konfiguracja
- 19.2. split brain
- 19.3. strategie deployowania
- 19.4. sposoby inicjalizacji
- 19.5. sharding
  - 19.5.1. sharding encji
  - 19.5.2. rebalancing

BOTTEGA IT minds