

Program szkolenia:

Budowanie skalowalnych aplikacji low latency w oparciu o Hazelcast in memory data grid

Informacje:

Nazwa:	Budowanie skalowalnych aplikacji low latency w oparciu o Hazelcast in memory data grid
Kod:	BigDataML-hazel
Kategoria:	BigData, streaming i Machine Learning
Grupa docelowa:	
Czas trwania:	2 dni
Forma:	20% wykłady / 30% warsztaty

W trakcie szkolenia nauczymy się jak zbudować rozproszony in-memory data grid używając technologii Hazelcast. Dzięki użyciu tej technologii będziemy mogli pisać efektywne przetwarzania streamingowe operujące na tysiącach eventów na sekundę. Będziemy mogli tworzyć aplikacje REST które są w stanie odpowiadać na requesty w mniej niż 10 ms!

W przetwarzaniach streamingowych największym problemem jest przechowywanie stanu. Aby osiągnąć low latency na ścieżce write round-trip do bazy danych jest zbyt długi. W aplikacjach REST często mierzymy się z problemem rozproszonego cache, tak aby obsługiwać klientów naszego API zachowując low latency, a jednocześnie odciążać bazy danych. Technologie typu in memory data grid pozwalają na rozwiązanie tych problemów.

Zalety szkolenia:

- Realne problemy dużej skali
- Zrozumienie pryncypiów
- Tworzenie własnych rozszerzeń

Szczegółowy program:

1. Czym jest Hazelcast?

- 1.1. Architektura Hazelcast
- 1.2. Kiedy używać
- 1.3. Use Cases
- 1.4. partycjonowanie danych w Hz

2. Setup Hazelcast

- 2.1. zrozumienie parametrów Hazelcast
- 2.2. Setup klastra składającego się z kilku nodów

3. Hazelcast DataTypes

- 3.1. zrozumienie struktury IMap
- 3.2. zrozumienie struktury near cache
- 3.3. pisanie własnego near cache używanego przez Hz

4. Monitorowanie Hz

- 4.1. zrozumienie metryk
- 4.2. optymalizacja odpowiednich wskaźników

5. Serializacja i Deserializacja danych w HZ

- 5.1. używanie custom kryto serializer
- 5.2. używanie system kryto deserialzier

6. Pisanie customowego persistence store

- 6.1. rzucanie danych w sposób asynchroniczny i sync
- 6.2. używanie Mongo Jako DataStore
- 6.3. konfiguracja TTL data-store

7. Pisanie własnego AbstractDiscoveryStrategy

7.1. stworzenie customowego discovery strategy

7.2. tworzenie mechanizmu do tego aby nody Hz widziały się nawzajem

8. Rolling Deployment Hz i Cold Start

8.1. hazelcast.initial.min.cluster.size parametr do odpowiedniego startu Cold

8.2. Optymalizacja procesu re-partitioningu