

Obsługa aplikacji, które operują na ogromnych zbiorach danych, czyli na przykład portali społecznościowych, przekracza możliwości zwykłych relacyjnych baz. Praca ze złożonymi zbiorami danych wymaga architektury obejmującej wielomaszynowe klastry, dzięki którym możliwe jest przechowywanie i przesyłanie informacji praktycznie dowolnej wielkości. Architektura taka powinna dodatkowo być prosta w użyciu, niezawodna i skalowalna.

Dzięki tej książce nauczysz się budować tego rodzaju architekturę. Zapoznasz się z technologią wykorzystywania klastrów maszyn. Dowiesz się, jak działają narzędzia przeznaczone specjalnie do przechwytywania i analizy danych na wielką skalę. W książce zaprezentowano łatwe do zrozumienia podejście do obsługi systemów wielkich zbiorów danych, które mogą być budowane i uruchamiane przez niewielki zespół. Nie zabrakło też wyczerpującego opisu praktycznej implementacji systemu **Big Data** z wykorzystaniem rzeczywistego przykładu.

W tej książce znajdziesz:

- teoretyczne podstawy koncepcji systemów **Big Data**
- wskazówki umożliwiające optymalne wykorzystanie zasobów do obsługi danych
- wybór technik przetwarzania i obsługi wielkich ilości danych w czasie rzeczywistym
- zagadnienia dotyczące baz danych **NoSQL**, przetwarzania strumieniowego i zarządzania złożonością obliczeń przyrostowych
- informacje o praktycznym stosowaniu takich narzędzi jak **Hadoop**, **Cassandra** i **Storm**
- wskazówki umożliwiające poszerzenie wiedzy o zwykłych bazach danych

## **Big Data – to skalowalność i prostota obsługi wielkich ilości danych!**

### **Przedmowa (11)**

### **Podziękowania (13)**

### **O książce (17)**

### **Rozdział 1. Nowy paradygmat dla Big Data (19)**

- 1.1. Zawartość książki (20)
- 1.2. Skalowanie tradycyjnej bazy danych (21)
  - 1.2.1. Skalowanie za pomocą kolejki (22)
  - 1.2.2. Skalowanie przez sharding bazy danych (22)
  - 1.2.3. Rozpoczynają się problemy z odpornością na błędy (23)
  - 1.2.4. Problemy z uszkodzeniem danych (24)
  - 1.2.5. Co poszło nie tak? (24)
  - 1.2.6. W jaki sposób techniki Big Data mogą pomóc? (24)
- 1.3. NoSQL nie jest panaceum (25)
- 1.4. Pierwsze zasady (25)
- 1.5. Wymagane właściwości systemu Big Data (26)
  - 1.5.1. niezawodność i odporność na błędy (26)
  - 1.5.2. Odczytywanie i aktualizowanie z niską latencją (27)
  - 1.5.3. Skalowalność (27)
  - 1.5.4. Uogólnienie (27)
  - 1.5.5. Rozszerzalność (27)
  - 1.5.6. Zapytania ad hoc (28)
  - 1.5.7. Minimalna konserwacja (28)
  - 1.5.8. Debugowalność (28)
- 1.6. Problemy z architektuрами w pełni przyrostowymi (29)
  - 1.6.1. Złożoność operacyjna (29)
  - 1.6.2. Ekstremalna złożoność osiągnięcia spójności ostatecznej (30)
  - 1.6.3. Brak odporności na ludzkie błędy (32)
  - 1.6.4. Rozwiązanie w pełni przyrostowe w porównaniu z architekturą lambda (32)

- 1.7. Architektura lambda (34)
  - 1.7.1. Warstwa przetwarzania wsadowego (36)
  - 1.7.2. Warstwa obsługująca (37)
  - 1.7.3. Warstwy przetwarzania wsadowego i obsługująca zapewniają niemal wszystkie właściwości (37)
  - 1.7.4. Warstwa przetwarzania czasu rzeczywistego (39)
- 1.8. Najnowsze trendy w technologii (41)
  - 1.8.1. Procesory nie stają się coraz szybsze (42)
  - 1.8.2. Elastyczne chmury (42)
  - 1.8.3. Dynamiczny ekosystem open source dla Big Data (42)
- 1.9. Przykładowa aplikacja: SuperWebAnalytics.com (44)
- 1.10. Podsumowanie (44)

## **CZĘŚĆ I. WARSTWA PRZETWARZANIA WSADOWEGO (47)**

### **Rozdział 2. Model danych dla Big Data (49)**

- 2.1. Właściwości danych (51)
  - 2.1.1. Dane są surowe (53)
  - 2.1.2. Dane są niemutowalne (56)
  - 2.1.3. Dane są wiecznie prawdziwe (59)
- 2.2. Reprezentacja danych za pomocą modelu opartego na faktach (60)
  - 2.2.1. Przykładowe fakty i ich właściwości (60)
  - 2.2.2. Korzyści ze stosowania modelu opartego na faktach (62)
- 2.3. Schematy graficzne (66)
  - 2.3.1. Elementy schematu graficznego (66)
  - 2.3.2. Potrzeba zapewnienia egzekwowalności schematu (67)
- 2.4. Kompletny model danych dla aplikacji SuperWebAnalytics.com (68)
- 2.5. Podsumowanie (70)

### **Rozdział 3. Model danych dla Big Data: ilustracja (71)**

- 3.1. Dlaczego framework serializacji? (72)
- 3.2. Apache Thrift (72)
  - 3.2.1. Węzły (73)
  - 3.2.2. Krawędzie (73)
  - 3.2.3. Właściwości (74)
  - 3.2.4. Połączenie wszystkich elementów w obiekty danych (75)
  - 3.2.5. Ewolucja schematu (75)
- 3.3. Ograniczenia frameworku serializacji (76)
- 3.4. Podsumowanie (78)

### **Rozdział 4. Przechowywanie danych w warstwie przetwarzania wsadowego (79)**

- 4.1. Wymagania dotyczące przechowywania głównego zbioru danych (80)
- 4.2. Wybór rozwiązania pamięci masowej dla warstwy przetwarzania wsadowego (81)
  - 4.2.1. Użycie magazynu danych klucz-wartość dla głównego zbioru danych (82)
  - 4.2.2. Rozproszone systemy plików (82)
- 4.3. Sposób działania rozproszonych systemów plików (83)
- 4.4. Przechowywanie głównego zbioru danych z wykorzystaniem rozproszonego systemu plików (85)
- 4.5. Partycjonowanie pionowe (86)
- 4.6. Niskopoziomowy charakter rozproszonych systemów plików (87)
- 4.7. Przechowywanie głównego zbioru danych aplikacji SuperWebAnalytics.com w rozproszonym systemie plików (89)

#### 4.8. Podsumowanie (90)

### **Rozdział 5. Przechowywanie danych w warstwie przetwarzania wsadowego: ilustracja (91)**

- 5.1. Korzystanie z Hadoop Distributed File System (92)
  - 5.1.1. Problem małych plików (93)
  - 5.1.2. Dążenie do wyższego poziomu abstrakcji (93)
- 5.2. Przechowywanie danych w warstwie przetwarzania wsadowego z wykorzystaniem biblioteki Pail (94)
  - 5.2.1. Podstawowe operacje biblioteki Pail (95)
  - 5.2.2. Serializacja i umieszczanie obiektów w wiaderkach (96)
  - 5.2.3. Operacje przetwarzania wsadowego z wykorzystaniem biblioteki Pail (98)
  - 5.2.4. Partycjonowanie pionowe z wykorzystaniem biblioteki Pail (99)
  - 5.2.5. Formaty plików i kompresja biblioteki Pail (100)
  - 5.2.6. Podsumowanie zalet biblioteki Pail (101)
- 5.3. Przechowywanie głównego zbioru danych dla aplikacji SuperWebAnalytics.com (102)
  - 5.3.1. Ustrukturyzowane wiaderko dla obiektów Thrift (103)
  - 5.3.2. Podstawowe wiaderko dla aplikacji SuperWebAnalytics.com (104)
  - 5.3.3. Podział wiaderka w celu pionowego partycjonowania zbioru danych (104)
- 5.4. Podsumowanie (107)

### **Rozdział 6. Warstwa przetwarzania wsadowego (109)**

- 6.1. Przykłady do rozważenia (110)
  - 6.1.1. Liczba odsłon w czasie (110)
  - 6.1.2. Inferencja płci (111)
  - 6.1.3. Punkty wpływu (111)
- 6.2. Obliczenia w warstwie przetwarzania wsadowego (112)
- 6.3. Porównanie algorytmów ponownego obliczania z algorytmami przyrostowymi (114)
  - 6.3.1. Wydajność (116)
  - 6.3.2. Odporność na ludzkie błędy (117)
  - 6.3.3. Ogólność algorytmów (117)
  - 6.3.4. Wybór stylu algorytmu (118)
- 6.4. Skalowalność w warstwie przetwarzania wsadowego (119)
- 6.5. MapReduce: paradygmat dla obliczeń Big Data (119)
  - 6.5.1. Skalowalność (121)
  - 6.5.2. Odporność na błędy (123)
  - 6.5.3. Ogólność MapReduce (123)
- 6.6. Niskopoziomowy charakter MapReduce (125)
  - 6.6.1. Wieloetapowe obliczenia są nienaturalne (125)
  - 6.6.2. Operacje łączenia są bardzo skomplikowane do ręcznej implementacji (126)
  - 6.6.3. Wykonywanie logiczne jest ściśle powiązane z fizycznym (128)
- 6.7. Diagramy potokowe: wyższy poziom sposobu myślenia na temat obliczeń wsadowych (129)
  - 6.7.1. Koncepcje diagramów potokowych (129)
  - 6.7.2. Wykonywanie diagramów potokowych poprzez MapReduce (134)
  - 6.7.3. Agregator łączący (134)
  - 6.7.4. Przykłady diagramów potokowych (136)
- 6.8. Podsumowanie (136)

### **Rozdział 7. Warstwa przetwarzania wsadowego: ilustracja (139)**

- 7.1. Przykład ilustracyjny (140)
- 7.2. Typowe pułapki narzędzi do przetwarzania danych (142)
  - 7.2.1. Języki niestandardowe (142)

- 7.2.2. Słabo komponowalne abstrakcje (143)
- 7.3. Wprowadzenie do JCasalog (144)
  - 7.3.1. Model danych JCasalog (144)
  - 7.3.2. Struktura zapytania JCasalog (145)
  - 7.3.3. Kwerendowanie wielu zbiorów danych (147)
  - 7.3.4. Grupowanie i agregatory (150)
  - 7.3.5. Analiza przykładowego zapytania (150)
  - 7.3.6. Niestandardowe operacje predykatów (153)
- 7.4. Kompozycja (158)
  - 7.4.1. Łączenie podzapytań (158)
  - 7.4.2. Podzapytania tworzone dynamicznie (159)
  - 7.4.3. Makra predykatów (162)
  - 7.4.4. Makra predykatów tworzone dynamicznie (164)
- 7.5. Podsumowanie (166)

## **Rozdział 8. Przykładowa warstwa przetwarzania wsadowego: architektura i algorytmy (167)**

- 8.1. Projekt warstwy przetwarzania wsadowego aplikacji SuperWebAnalytics.com (168)
  - 8.1.1. Obsługiwane zapytania (168)
  - 8.1.2. Obrazy wsadowe (169)
- 8.2. Przegląd przepływu pracy (172)
- 8.3. Przyjmowanie nowych danych (174)
- 8.4. Normalizacja adresów URL (174)
- 8.5. Normalizacja identyfikatorów użytkowników (175)
- 8.6. Usuwanie zduplikowanych odstępów (180)
- 8.7. Obliczanie obrazów wsadowych (180)
  - 8.7.1. Liczba odstępów w czasie (180)
  - 8.7.2. Liczba unikatowych użytkowników w czasie (181)
  - 8.7.3. Analiza współczynnika odrzuceń (182)
- 8.8. Podsumowanie (183)

## **Rozdział 9. Przykładowa warstwa przetwarzania wsadowego: implementacja (185)**

- 9.1. Punkt startowy (186)
- 9.2. Przygotowanie przepływu pracy (187)
- 9.3. Przyjmowanie nowych danych (187)
- 9.4. Normalizacja adresów URL (191)
- 9.5. Normalizacja identyfikatorów użytkowników (192)
- 9.6. Usuwanie zduplikowanych odstępów (197)
- 9.7. Obliczanie obrazów wsadowych (197)
  - 9.7.1. Liczba odstępów w czasie (197)
  - 9.7.2. Liczba unikatowych użytkowników w czasie (200)
  - 9.7.3. Analiza współczynnika odrzuceń (201)
- 9.8. Podsumowanie (204)

## **CZĘŚĆ II. WARSTWA OBSŁUGUJĄCA (205)**

### **Rozdział 10. Warstwa obsługująca (207)**

- 10.1. Metryki wydajności dla warstwy obsługującej (209)
- 10.2. Rozwiązanie warstwy obsługującej dotyczące problemu wyboru między normalizacją a denormalizacją (211)
- 10.3. Wymagania względem bazy danych warstwy obsługującej (213)
- 10.4. Projektowanie warstwy obsługującej dla aplikacji SuperWebAnalytics.com (215)
  - 10.4.1. Liczba odstępów w czasie (215)

- 10.4.2. Liczba użytkowników w czasie (216)
- 10.4.3. Analiza współczynnika odrzuceń (217)
- 10.5. Porównanie z rozwiązaniem w pełni przyrostowym (217)
  - 10.5.1. W pełni przyrostowe rozwiązanie problemu liczby unikatowych użytkowników w czasie (218)
  - 10.5.2. Porównanie z rozwiązaniem opartym na architekturze lambda (224)
- 10.6. Podsumowanie (224)

## **Rozdział 11. Warstwa obsługująca: ilustracja (227)**

- 11.1. Podstawy ElephantDB (228)
  - 11.1.1. Tworzenie obrazu w ElephantDB (228)
  - 11.1.2. Serwowanie obrazu w ElephantDB (229)
  - 11.1.3. Korzystanie z ElephantDB (229)
- 11.2. Budowanie warstwy obsługującej dla aplikacji SuperWebAnalytics.com (231)
  - 11.2.1. Liczba odsłon w czasie (231)
  - 11.2.2. Liczba unikatowych użytkowników w czasie (234)
  - 11.2.3. Analiza współczynnika odrzuceń (235)
- 11.3. Podsumowanie (236)

## **CZĘŚĆ III. WARSTWA PRZETWARZANIA CZASU RZECZYWISTEGO (237)**

### **Rozdział 12. Obrazy czasu rzeczywistego (239)**

- 12.1. Obliczanie obrazów czasu rzeczywistego (241)
- 12.2. Przechowywanie obrazów czasu rzeczywistego (242)
  - 12.2.1. Dokładność ostateczna (243)
  - 12.2.2. Ilość stanu przechowywanego w warstwie przetwarzania czasu rzeczywistego (244)
- 12.3. Wyzwania obliczeń przyrostowych (245)
  - 12.3.1. Słuszność twierdzenia CAP (245)
  - 12.3.2. Kompleksowa interakcja między twierdzeniem CAP a algorytmami przyrostowymi (247)
- 12.4. Porównanie aktualizacji asynchronicznych z synchronicznymi (249)
- 12.5. Wygaszanie obrazów czasu rzeczywistego (250)
- 12.6. Podsumowanie (253)

### **Rozdział 13. Obrazy czasu rzeczywistego: ilustracja (255)**

- 13.1. Model danych Cassandra (256)
- 13.2. Korzystanie z bazy danych Cassandra (257)
  - 13.2.1. Zaawansowane funkcje Cassandra (259)
- 13.3. Podsumowanie (259)

### **Rozdział 14. Kolejowanie i przetwarzanie strumieniowe (261)**

- 14.1. Kolejowanie (262)
  - 14.1.1. Serwery kolejek pojedynczego konsumenta (263)
  - 14.1.2. Kolejki wielu konsumentów (264)
- 14.2. Przetwarzanie strumieniowe (265)
  - 14.2.1. Kolejki i procesy robocze (266)
  - 14.2.2. Pułapki paradygmatu "kolejki i procesy robocze" (267)
- 14.3. Pojedyncze przetwarzanie strumieniowe wyższego poziomu (268)
  - 14.3.1. Model Storm (268)
  - 14.3.2. Zapewnianie przetwarzania komunikatów (272)
- 14.4. Warstwa przetwarzania czasu rzeczywistego dla aplikacji SuperWebAnalytics.com (274)
  - 14.4.1. Struktura topologii (277)

14.5. Podsumowanie (278)

## **Rozdział 15. Kolejowanie i przetwarzanie strumieniowe: ilustracja (281)**

- 15.1. Definiowanie topologii za pomocą Apache Storm (281)
- 15.2. Klastry Apache Storm i wdrażanie topologii (284)
- 15.3. Gwarantowanie przetwarzania komunikatów (286)
- 15.4. Implementacja warstwy przetwarzania czasu rzeczywistego aplikacji SuperWebAnalytics.com dla liczby unikatowych użytkowników w czasie (288)
- 15.5. Podsumowanie (292)

## **Rozdział 16. Mikrosadowe przetwarzanie strumieniowe (293)**

- 16.1. Osiąganie semantyki "dokładnie raz" (294)
  - 16.1.1. Ścisłe uporządkowane przetwarzanie (294)
  - 16.1.2. Mikrosadowe przetwarzanie strumieniowe (295)
  - 16.1.3. Topologie przetwarzania mikrosadowego (296)
- 16.2. Podstawowe koncepcje mikrosadowego przetwarzania strumieniowego (299)
- 16.3. Rozszerzanie diagramów potokowych dla przetwarzania mikrosadowego (300)
- 16.4. Dokończenie warstwy przetwarzania czasu rzeczywistego dla aplikacji SuperWebAnalytics.com (302)
  - 16.4.1. Liczba odsłon w czasie (302)
  - 16.4.2. Analiza współczynnika odrzuceń (302)
- 16.5. Inne spojrzenie na przykład analizy współczynnika odrzuceń (307)
- 16.6. Podsumowanie (308)

## **Rozdział 17. Mikrosadowe przetwarzanie strumieniowe: ilustracja (309)**

- 17.1. Korzystanie z interfejsu Trident (310)
- 17.2. Dokończenie warstwy przetwarzania czasu rzeczywistego dla aplikacji SuperWebAnalytics.com (313)
  - 17.2.1. Liczba odsłon w czasie (314)
  - 17.2.2. Analiza współczynnika odrzuceń (316)
- 17.3. W pełni odporne na błędy przetwarzanie mikrosadowe z utrzymywaniem stanu w pamięci (322)
- 17.4. Podsumowanie (323)

## **Rozdział 18. Tajniki architektury lambda (325)**

- 18.1. Definiowanie systemów danych (325)
- 18.2. Warstwa przetwarzania wsadowego i warstwa obsługująca (327)
  - 18.2.1. Przyrostowe przetwarzanie wsadowe (328)
  - 18.2.2. Pomiar i optymalizacja wykorzystania zasobów przez warstwę przetwarzania wsadowego (335)
- 18.3. Warstwa przetwarzania czasu rzeczywistego (339)
- 18.4. Warstwa zapytań (340)
- 18.5. Podsumowanie (341)

## **Skorowidz (343)**