

Spis treści

Przedmowa (9)

1. Wprowadzenie (13)

Krótką historia rozwoju systemów (14)

Krótką historia wzorców w rozwoju oprogramowania (15)

Formalizacja programowania algorytmicznego (15)

Wzorce programowania obiektowego (16)

Rozwój otwartego oprogramowania (16)

Wartość wzorców, praktyk i komponentów (17)

Stojąc na ramionach gigantów (17)

Wspólny język do dyskusji na temat naszych praktyk (18)

Współdzielone komponenty do łatwego ponownego wykorzystania (19)

Podsumowanie (20)

I. WZORCE JEDNOWĘZŁOWE (21)

2. Wzorzec Przyczepa (25)

Przykład przyczepy: dodawanie HTTPS do starszej usługi (26)

Dynamiczna konfiguracja za pomocą przyczepy (27)

Modułowe kontenery aplikacji (29)

Część praktyczna: wdrażanie kontenera topz (30)

Budowanie prostej usługi PaaS za pomocą przyczepy (31)

Projektowanie przyczep pod kątem modułowości i ponownego użycia (32)

Parametryzacja kontenerów (33)

Definiowanie API każdego kontenera (34)

Dokumentowanie kontenerów (35)

Podsumowanie (36)

3. Wzorzec Ambasador (37)

Używanie ambasadora do fragmentowania usługi (38)

Część praktyczna: implementacja pofragmentowanej usługi Redis (40)

Używanie ambasadora do pośredniczenia między usługami (42)

Używanie ambasadora do eksperymentowania lub rozdzielania żądań (43)

Część praktyczna: implementacja 10% eksperymentów (44)

4. Wzorzec Adapter (47)

Monitorowanie (48)

Część praktyczna: monitorowanie za pomocą systemu Prometheus (49)

Rejestrowanie (50)

Część praktyczna: normalizowanie różnych formatów rejestrowania za pomocą fluentd (52)

Dodawanie monitora poprawności działania (53)

Część praktyczna: dodawanie wszechstronnego monitorowania kondycji MySQL (54)

II. WZORCE SERWOWANIA USŁUG (57)

5. Zreplikowane usługi o zrównoważonym obciążeniu (61)

Usługi bezstanowe (61)

Sondy gotowości dla mechanizmu równoważenia obciążenia (63)

Część praktyczna: tworzenie zreplikowanej usługi w Kubernetes (63)

Usługi ze śledzeniem sesji (65)

Zreplikowane usługi warstwy aplikacji (67)

Wprowadzenie warstwy buforowania (67)

Wdrażanie pamięci podręcznej (68)

Część praktyczna: wdrażanie warstwy buforowania (69)

Rozszerzanie warstwy buforowania (72)

Ograniczanie przepustowości i obrona przed atakiem DoS (72)

Przerywanie połączenia SSL (73)

Część praktyczna: wdrażanie serwera nginx i przerywania połączenia SSL (74)

Podsumowanie (76)

6. Usługi pofragmentowane (77)

Pofragmentowane buforowanie (78)

Dlaczego możesz potrzebować pofragmentowanej pamięci podręcznej? (79)

Znaczenie pamięci podręcznej dla wydajności systemu (79)

Zreplikowane pofragmentowane pamięci podręczne (81)

Część praktyczna: wdrożenie ambasadora i systemu memcached dla pofragmentowanej pamięci podręcznej (82)

Funkcja fragmentująca (86)

Wybór klucza (87)

Spójne funkcje haszujące (89)

Część praktyczna: budowanie spójnego fragmentującego pośrednika HTTP (90)

Pofragmentowane zreplikowane serwowanie usług (91)

Systemy fragmentowania na gorąco (91)

7. Wzorzec Rozrzucaj-Zbieraj (93)

Wzorzec Rozrzucaj-Zbieraj z węzłem głównym jako dystrybutorem (94)

Część praktyczna: rozproszone wyszukiwanie dokumentów (95)

Rozrzucaj-Zbieraj z fragmentowaniem liści (96)

Część praktyczna: pofragmentowane wyszukiwanie dokumentów (97)

Wybieranie odpowiedniej liczby liści (98)

Skalowanie wzorca Rozrzucaj-Zbieraj pod kątem niezawodności i skali obliczeniowej (100)

8. Funkcje i przetwarzanie oparte na zdarzeniach (103)

Kiedy FaaS ma sens (104)

Zalety FaaS (104)

Wyzwania FaaS (105)

Potrzeba przetwarzania w tle (106)

Potrzeba przechowywania danych w pamięci (106)

Koszty ciągłego przetwarzania opartego na żądaniach (107)

Wzorce dla usług FaaS (107)

Wzorzec Dekorator: transformacja żądań lub odpowiedzi (107)

Część praktyczna: ustawianie wartości domyślnych żądania przed jego przetworzeniem (109)

Obsługa zdarzeń (110)

Część praktyczna: implementowanie uwierzytelniania dwuetapowego (111)

Potoki oparte na zdarzeniach (113)

Część praktyczna: implementowanie potoku w celu rejestracji nowego użytkownika (114)

9. Wybór własności (117)

Czy musisz wybierać węzeł główny? (119)

Podstawy wyboru węzła głównego (120)

Część praktyczna: wdrażanie etcd (122)

Implementacja blokad (123)

Część praktyczna: implementowanie blokad w etcd (126)

Implementowanie własności (127)

Część praktyczna: implementowanie dzierżaw w etcd (128)

Obsługa jednoczesnej manipulacji danymi (129)

III. WZORCE PRZETWARZANIA WSADOWEGO (133)

10. Systemy kolejek roboczych (135)

Ogólny system kolejki roboczej (135)

Interfejs kontenera źródłowego (136)

Interfejs kontenera roboczego (139)

Infrastruktura współdzielonej kolejki roboczej (140)

Część praktyczna: implementacja generowania miniaturk plików wideo (143)

Dynamiczne skalowanie węzłów roboczych (144)

Wzorzec Wiele Węzłów Roboczych (146)

11. Przetwarzanie wsadowe oparte na zdarzeniach (149)

Wzorce przetwarzania opartego na zdarzeniach (150)

Kopiarka (151)

Filtr (152)

Rozdzielacz (153)

Fragmentator (154)

Scalanie (156)

Część praktyczna: budowanie przepływu opartego na zdarzeniach dla rejestracji nowego użytkownika (157)

Infrastruktura "publikuj-subskrybuj" (159)

Część praktyczna: wdrażanie Kafki (159)

12. Skoordynowane przetwarzanie wsadowe (163)

Łączenie (czyli synchronizacja barierowa) (163)

Redukcja (166)

Część praktyczna: zliczanie (166)

Suma (167)

Histogram (168)

Część praktyczna: znakowanie obrazów i potok przetwarzania (169)

13. Wniosek: nowy początek? (173)

Skorowidz (175)