

Przedmowa (9)

1. Wydajny kod Python (15)

- Podstawowy system komputerowy (15)
 - o Jednostki obliczeniowe (16)
 - o Jednostki pamięci (19)
 - o Warstwy komunikacji (21)
- Łączenie ze sobą podstawowych elementów (22)
 - o Porównanie wyidealizowanego przetwarzania z maszyną wirtualną języka Python (23)
- Dlaczego warto używać języka Python? (26)

2. Użycie profilowania do znajdowania wąskich gardeł (29)

- Efektywne profilowanie (30)
- Wprowadzenie do zbioru Julii (31)
- Obliczanie pełnego zbioru Julii (34)
- Proste metody pomiaru czasu - instrukcja print i dekorator (37)
- Prosty pomiar czasu za pomocą polecenia time systemu Unix (40)
- Użycie modułu cProfile (41)
- Użycie narzędzia runsnake do wizualizacji danych wyjściowych modułu cProfile (46)
- Użycie narzędzia line_profiler do pomiarów dotyczących kolejnych wierszy kodu (46)
- Użycie narzędzia memory_profiler do diagnozowania wykorzystania pamięci (51)
- Inspekcja obiektów w stercie za pomocą narzędzia heapy (56)
- Użycie narzędzia dowser do generowania aktywnego wykresu dla zmiennych z utworzonymi instancjami (58)
- Użycie modułu dis do sprawdzania kodu bajtowego narzędzia CPython (60)
 - o Różne metody, różna złożoność (62)
- Testowanie jednostkowe podczas optymalizacji w celu zachowania poprawności (64)
 - o Dekorator @profile bez operacji (64)
- Strategie udanego profilowania kodu (66)
- Podsumowanie (67)

3. Listy i krotki (69)

- Bardziej efektywne wyszukiwanie (71)
- Porównanie list i krotek (73)
- Listy jako tablice dynamiczne (74)
- Krotki w roli tablic statycznych (77)
- Podsumowanie (78)

4. Słowniki i zbiory (79)

- Jak działają słowniki i zbiory? (82)
 - o Wstawianie i pobieranie (82)
 - o Usuwanie (85)
 - o Zmiana wielkości (85)
 - o Funkcje mieszania i entropia (86)
- Słowniki i przestrzenie nazw (89)
- Podsumowanie (92)

5. Iteratory i generatory (93)

- Iteratory dla szeregów nieskończonych (96)
- Wartościowanie leniwe generatora (97)
- Podsumowanie (101)

6. Obliczenia macierzowe i wektorowe (103)

- Wprowadzenie do problemu (104)
- Czy listy języka Python są wystarczająco dobre? (107)
 - Problemy z przesadną alokacją (109)
- Fragmentacja pamięci (111)
 - Narzędzie perf (113)
 - Podejmowanie decyzji z wykorzystaniem danych wyjściowych narzędzia perf (115)
 - Wprowadzenie do narzędzia numpy (116)
- Zastosowanie narzędzia numpy w przypadku problemu dotyczącego dyfuzji (119)
 - Przydziały pamięci i operacje wewnętrzne (121)
 - Optymalizacje selektywne: znajdowanie tego, co wymaga poprawienia (124)
- Moduł numexpr: przyspieszanie i upraszczanie operacji wewnętrznych (127)
- Przestroga: weryfikowanie "optymalizacji" (biblioteka scipy) (129)
- Podsumowanie (131)

7. Kompilowanie do postaci kodu C (133)

- Jakie wzrosty szybkości są możliwe? (134)
- Porównanie kompilatorów JIT i AOT (136)
- Dlaczego informacje o typie ułatwiają przyspieszenie działania kodu? (136)
- Użycie kompilatora kodu C (137)
- Analiza przykładu zbioru Julii (138)
- Cython (139)
 - Kompilowanie czystego kodu Python za pomocą narzędzia Cython (139)
 - Użycie adnotacji kompilatora Cython do analizowania bloku kodu (141)
 - Dodawanie adnotacji typu (143)
- Shed Skin (147)
 - Tworzenie modułu rozszerzenia (148)
 - Koszt związany z kopiami pamięci (150)
- Cython i numpy (151)
 - Przetwarzanie równoległe rozwiązania na jednym komputerze z wykorzystaniem interfejsu OpenMP (152)
- Numba (154)
- Pythran (155)
- PyPy (157)
 - Różnice związane z czyszczeniem pamięci (158)
 - Uruchamianie interpretera PyPy i instalowanie modułów (159)
- Kiedy stosować poszczególne technologie? (160)
 - Inne przyszłe projekty (162)
 - Uwaga dotycząca układów GPU (162)
 - Oczekiwania dotyczące przyszłego projektu kompilatora (163)
- Interfejsy funkcji zewnętrznych (163)

- o ctypes (164)
 - o cffi (166)
 - o f2py (169)
 - o Moduł narzędzia CPython (171)
- Podsumowanie (174)

8. Współbieżność (175)

- Wprowadzenie do programowania asynchronicznego (176)
- Przeszukiwacz szeregowy (179)
- gevent (181)
- tornado (185)
- AsyncIO (188)
- Przykład z bazą danych (190)
- Podsumowanie (193)

9. Moduł multiprocessing (195)

- Moduł multiprocessing (198)
- Przybliżenie liczby pi przy użyciu metody Monte Carlo (200)
- Przybliżanie liczby pi za pomocą procesów i wątków (201)
 - o Zastosowanie obiektów języka Python (201)
 - o Liczby losowe w systemach przetwarzania równoległego (208)
 - o Zastosowanie narzędzia numpy (209)
- Znajdowanie liczb pierwszych (211)
 - o Kolejki zadań roboczych (217)
- Weryfikowanie liczb pierwszych za pomocą komunikacji międzyprocesowej (221)
 - o Rozwiązanie z przetwarzaniem szeregowym (225)
 - o Rozwiązanie z prostym obiektem Pool (225)
 - o Rozwiązanie z bardzo prostym obiektem Pool dla mniejszych liczb (227)
 - o Użycie obiektu Manager.Value jako flagi (228)
 - o Użycie systemu Redis jako flagi (229)
 - o Użycie obiektu RawValue jako flagi (232)
 - o Użycie modułu mmap jako flagi (232)
 - o Użycie modułu mmap do odtworzenia flagi (233)
- Współużytkowanie danych narzędzia numpy za pomocą modułu multiprocessing (236)
- Synchronizowanie dostępu do zmiennych i plików (242)
 - o Blokowanie plików (242)
 - o Blokowanie obiektu Value (245)
- Podsumowanie (248)

10. Klastry i kolejki zadań (249)

- Zalety klastrowania (250)
- Wady klastrowania (251)
 - o Strata o wartości 462 milionów dolarów na giełdzie Wall Street z powodu kiepskiej strategii aktualizacji klastra (252)
 - o 24-godzinny przestój usługi Skype w skali globalnej (253)
- Typowe projekty klastrowe (254)

- Metoda rozpoczęcia tworzenia rozwiązania klastrowego (254)
- Sposoby na uniknięcie kłopotów podczas korzystania z klastrów (255)
- Trzy rozwiązania klastrowe (257)
 - o Użycie modułu Parallel Python dla prostych klastrów lokalnych (257)
 - o Użycie modułu IPython Parallel do obsługi badań (259)
- Użycie systemu NSQ dla niezawodnych klastrów produkcyjnych (262)
 - o Kolejki (263)
 - o Publikator/subskrybent (264)
 - o Rozproszone obliczenia liczb pierwszych (266)
- Inne warte uwagi narzędzia klastrowania (268)
- Podsumowanie (269)

11. Mniejsze wykorzystanie pamięci RAM (271)

- Obiekty typów podstawowych są kosztowne (272)
 - o Moduł array zużywa mniej pamięci do przechowywania wielu obiektów typu podstawowego (273)
- Analiza wykorzystania pamięci RAM w kolekcji (276)
- Bajty i obiekty Unicode (277)
- Efektywne przechowywanie zbiorów tekstowych w pamięci RAM (279)
 - o Zastosowanie metod dla 8 milionów tokenów (280)
- Wskazówki dotyczące mniejszego wykorzystania pamięci RAM (288)
- Probabilistyczne struktury danych (289)
 - o Obliczenia o bardzo dużym stopniu przybliżenia z wykorzystaniem jednobajтового licznika Morrisa (290)
 - o Wartości k-minimum (291)
 - o Filtry Blooma (295)
 - o Licznik LogLog (299)
 - o Praktyczny przykład (303)

12. Rady specjalistów z branży (307)

- Narzędzie Social Media Analytics (SoMA) firmy Adaptive Lab (307)
 - o Język Python w firmie Adaptive Lab (308)
 - o Projekt narzędzia SoMA (308)
 - o Zastosowana metodologia projektowa (309)
 - o Serwisowanie systemu SoMA (309)
 - o Rada dla inżynierów z branży (310)
- Technika głębokiego uczenia prezentowana przez firmę RadimRehurek.com (310)
 - o Strzał w dziesiątkę (311)
 - o Rady dotyczące optymalizacji (313)
 - o Podsumowanie (315)
- Uczenie maszynowe o dużej skali gotowe do zastosowań produkcyjnych w firmie Lyst.com (315)
 - o Rola języka Python w witrynie Lyst (316)
 - o Projekt klastra (316)
 - o Ewolucja kodu w szybko rozwijającej się nowej firmie (316)
 - o Budowanie mechanizmu rekomendacji (316)
 - o Raportowanie i monitorowanie (317)
 - o Rada (317)

- Analiza serwisu społecznościowego o dużej skali w firmie Smesh (318)
 - o Rola języka Python w firmie Smesh (318)
 - o Platforma (318)
 - o Dopasowywanie łańcuchów w czasie rzeczywistym z dużą wydajnością (319)
 - o Raportowanie, monitorowanie, debugowanie i wdrażanie (320)
- Interpreter PyPy zapewniający powodzenie systemów przetwarzania danych i systemów internetowych (322)
 - o Wymagania wstępne (322)
 - o Baza danych (323)
 - o Aplikacja internetowa (323)
 - o Mechanizm OCR i tłumaczenie (324)
 - o Dystrybucja zadań i procesy robocze (324)
 - o Podsumowanie (325)
- Kolejki zadań w serwisie internetowym Lanyrd.com (325)
 - o Rola języka Python w serwisie Lanyrd (325)
 - o Zapewnianie odpowiedniej wydajności kolejki zadań (326)
 - o Raportowanie, monitorowanie, debugowanie i wdrażanie (326)
 - o Rada dla programistów z branży (326)

Skorowidz (329)