BO·TT·EGA
IT minds

Program szkolenia:

Wydajny Python

Informacje:

Nazwa:	**Wydajny Python**
Kod:	**python-perf**
Kategoria:	Python
Odbiorcy:	developerzy, architekci
Czas trwania:	3 dni
Forma:	30% wykłady / 70% warsztaty

Szkolenie z poprawiania wydajności w projektach opartych o Pythona. Bezlitośnie porusza słabe strony języka i uczy je obchodzić. Uczestnik po ukończeniu będzie umiał znaleźć źródła problemów wydajnościowych i odpowiednio spriorytetyzować ich usunięcie.

Dzień 1: Jak mierzyć performance w Pythonie, czyli szukanie właściwego problemu do rozwiązania

Dzień 2: Optymalizowanie na poziomie kodu

Dzień 3: Optymalizowanie na poziomie architektury

Zalety szkolenia:

- Studia przypadku
- Systemy rozproszone
- Narzędzia i pułapki

BO·TT·EGA
IT minds

Szczegółowy program:

BOTTEGA IT minds

6. Optymalizowanie przy problemie z bazą danych

- 6.1. Profilowanie i optymalizacja zapytań na przykładzie PostgreSQL
- 6.2. Cache'owanie
  - 6.2.1. Jak przebudowywać cache?
- 6.3. Tworzenie dedykowanego modelu do odczytu (ang. read model)
  - 6.3.1. Synchronicznie
  - 6.3.2. Asynchronicznie

7. Dobieranie struktury danych do problemu

- 7.1. Lista
- 7.2. Krotka (ang. tuple)
- 7.3. Słownik
- 7.4. Zbiór
- 7.5. collections.deque
- 7.6. Czy ma sens implementowanie swojej struktury danych?

8. Analizowanie wolnego kodu

- 8.1. Notacja dużego O - przypadki pesymistyczne
- 8.2. Szukanie dominującej operacji
- 8.3. Zastępowanie szeregu instrukcji warunkowych
  - 8.3.1. słownikiem
  - 8.3.2. polimorfizmem

9. Przetwarzanie równoległe i asynchroniczne

- 9.1. Jak zrównoleglić obliczenia?
  - 9.1.1. Nie każdy problem da się zrównoleglić
  - 9.1.2. Pula wątków - ThreadPoolExecutor
  - 9.1.3. Pula procesów - ProcessPoolExecutor

BOTTEGA IT minds

- 9.2. Dask
- 9.3. Kolejki zadań
  - 9.3.1. Tworzenie szeregu zadań
  - 9.3.2. Wzorzec Polling + puszczenie zadanie do kolejki

10. Wydajność w systemach rozproszonych

- 10.1. Najpierw zmierz
  - 10.1.1. Distributed tracing z wykorzystaniem zewnętrznych usług
  - 10.1.2. Distributed tracing z wykorzystaniem Zipkina
- 10.2. Znajdowanie wąskich gardeł i ich optymalizacja
- 10.3. Budowanie metryk

11. Na problemy klasy IO-bound - Asyncio

- 11.1. C10k problem
- 11.2. Kiedy używać
- 11.3. Modelowanie z użyciem asyncio
- 11.4. Testowanie

12. Studia przypadków

- 12.1. Czat
- 12.2. Web scraping
- 12.3. Aukcje internetowe
  - 12.3.1. tradycyjna aukcja
  - 12.3.2. ślepa aukcja
- 12.4. Silnik giełdowy z obsługą zleceń LIMIT i STOP LOSS
- 12.5. Python z JITem - PyPy
- 12.6. Cython
  - 12.6.1. Podstawy

BO·TT·EGA
IT minds

12.6.2. Optymalizujemy kompilując wybrane moduły