

Использование библиотеки Bokeh для визуализации данных процессоров

Гиренко Александр Евгеньевич

Приамурский государственный университет им Шолом-Алейхема

Студент

Аннотация

В данной статье рассматривается применение библиотеки Bokeh для визуализации данных о производительности процессоров. На основе данных из файла CPU_Efficiency.csv созданы интерактивные графики, позволяющие анализировать зависимости между различными характеристиками процессоров, такими как количество ядер, потоков, базовая и турбо-частота, а также результаты тестов производительности (singleScore и multiScore). Продемонстрированы возможности библиотеки Bokeh для создания интерактивных визуализаций, которые могут быть использованы для анализа и интерпретации данных.

Ключевые слова: Bokeh, визуализация данных, процессоры, производительность, интерактивные графики.

Using the bokeh library for visualizing processor data

Girenko Aleksandr Evgen'evich

Sholom-Aleichem Priamursky State University

Student

Abstract

This article discusses the use of the Bokeh library for visualizing data on processor performance. Based on data from the CPU_Efficiency.csv file, interactive graphs were created to analyze the relationships between various processor characteristics, such as core count, thread count, base and turbo frequencies, as well as performance test results (singleScore and multiScore). The capabilities of the Bokeh library for creating interactive visualizations that can be used for data analysis and interpretation are demonstrated.

Keywords: Bokeh, data visualization, processors, performance, interactive graphs.

1-Введение

1.1 Актуальность

В современном мире анализ данных играет ключевую роль в принятии решений. Визуализация данных позволяет наглядно представить сложные зависимости и выявить закономерности. Библиотека Bokeh предоставляет мощные инструменты для создания интерактивных графиков, которые могут быть использованы для анализа данных о производительности процессоров.

Это особенно актуально для специалистов в области компьютерных технологий, которые работают с большими объемами данных.

1.2 Обзор исследований

У. Маккини рассмотрел методы анализа данных с использованием Python, включая работу с библиотеками Pandas и NumPy, которые часто применяются вместе с Bokeh для визуализации данных [1]. Дж. ВандерПлас. Подробно изучил инструменты для работы с данными в Python, включая визуализацию с использованием библиотек Matplotlib, Seaborn и Bokeh [2]. М. Гринберг описал разработку веб-приложений на Python с использованием Flask, а также интеграцию Bokeh для создания интерактивных веб-приложений [3]. А.С. Хан разработал масштабируемый и интерактивный веб-инструмент для исследования данных JRO с использованием библиотеки Bokeh на Python [4]. Ч. Чай и др. исследовали интерактивную визуализацию сложных сейсмических данных и моделей с использованием библиотеки Bokeh [5].

1.3 Цель исследования

Цель исследования — создать интерактивные графики для визуализации данных о производительности процессоров с использованием библиотеки Bokeh и проанализировать зависимости между различными параметрами.

2. Материалы и Методы

2.1. Исходные данные

Для анализа использовался файл CPU_Efficiency.csv, содержащий информацию о характеристиках процессоров, таких как количество ядер, потоков, базовая и турбо-частота, а также результаты тестов производительности (singleScore и multiScore). Данные были загружены в среду Python с использованием библиотеки Pandas. Файл можно скачать по ссылке: <https://www.kaggle.com/datasets/alanjo/cpu-benchmarks/data>

2.2. Используемые технологии

Для визуализации данных использовалась библиотека Bokeh которая позволяет создавать интерактивные графики с поддержкой инструментов наведения и масштабирования. Она поддерживает различные типы графиков, такие как scatter plots, line plots и bar charts.

2.3. Методы визуализации

Для анализа данных были созданы следующие графики:

1. Зависимость singleScore от multiScore.
2. Зависимость singleScore от количества ядер.
3. Зависимость multiScore от количества потоков.
4. Зависимость singleScore от базовой частоты.
5. Зависимость multiScore от турбо-частоты.

6. Зависимость singleScore от типа процессора (например, Desktop или Laptop).

3. Программная реализация

Для визуализации данных о производительности процессоров была использована библиотека Bokeh, которая позволяет создавать интерактивные графики. Ниже приведен код, реализующий создание шести графиков для анализа зависимостей между различными характеристиками процессоров.

3.1 Импорт необходимых библиотек и модулей

Для реализации визуализации данных были использованы следующие библиотеки и модули: (рис.1).

```
import pandas as pd
from bokeh.plotting import figure, show
from bokeh.io import output_file
from bokeh.models import ColumnDataSource, HoverTool
from bokeh.layouts import gridplot
from bokeh.transform import factor_cmap
from bokeh.palettes import Spectral6
import os
```

Рисунок 1- Импорт необходимых библиотек и модулей для визуализации данных с использованием Bokeh

Pandas— это библиотека для анализа и обработки данных на языке Python. Она предоставляет мощные инструменты для работы с табличными данными, такие как DataFrame, которые позволяют легко загружать, фильтровать, сортировать и анализировать данные из CSV-файлов и других источников.

Bokeh — это библиотека для создания интерактивных визуализаций на языке Python. Она позволяет строить графики, диаграммы и другие визуализации, которые можно легко встраивать в веб-приложения. Bokeh поддерживает множество типов графиков и инструментов для взаимодействия с данными, таких как масштабирование, наведение и панорамирование.

os — это встроенный модуль Python, который предоставляет функции для взаимодействия с операционной системой. В данном случае он используется для проверки существования файла перед его загрузкой, что помогает избежать ошибок, связанных с отсутствием файла.

3.2 Загрузка данных

Если файл существует, данные загружаются в DataFrame с помощью pd.read_csv(). Если файл отсутствует, выводится сообщение об ошибке (рис.2).

```
file_path = r"C:\girenko\CPU_Efficiency.csv" # Указание пути к файлу

if not os.path.exists(file_path):
    print(f"Файл {file_path} не найден! Убедитесь, что файл находится в правильной папке.")
else:

    df = pd.read_csv(file_path)
```

Рисунок 2- Загрузка данных из CSV-файла и преобразование в ColumnDataSource для Bokeh

3.3 Подготовка данных для визуализации

Для работы с библиотекой Bokeh данные необходимо преобразовать в объект ColumnDataSource.(рис.3)

```
df = pd.read_csv(file_path)

source = ColumnDataSource(df)
```

Рисунок 3- Загрузка данных и подготовка для визуализации в Bokeh

3.4 Настройка вывода

Для сохранения графиков в HTML-файл используется функция `output_file`. Для сохранения созданных графиков в интерактивном формате используется функция `output_file` из библиотеки Bokeh. В данном случае результат визуализации сохраняется в файл с именем `cpu_visualization.html`. Этот файл можно открыть в любом веб-браузере, чтобы просмотреть интерактивные графики, которые позволяют масштабировать, перемещаться по данным и получать дополнительную информацию при наведении на точки.(рис.4).

```
output_file("cpu_visualization.html")
```

Рисунок 4 - Настройка вывода графиков в HTML-файл с использованием Bokeh

3.5 Создание графиков

```

p1.add_tools(hover)

p2 = figure(title="Зависимость singleScore от количества ядер", width=600, height=400,
            x_axis_label="Количество ядер", y_axis_label="singleScore")
p2.scatter('cores', 'singleScore', source=source, size=8, color="green", alpha=0.6)

p2.add_tools(HoverTool(tooltips=[
    ("Процессор", "@cpuName"),
    ("Производитель", "@manufacturer"),
    ("Потоки", "@threads")
]))

p3 = figure(title="Зависимость multiScore от количества потоков", width=600, height=400,
            x_axis_label="Количество потоков", y_axis_label="multiScore")
p3.scatter('threads', 'multiScore', source=source, size=8, color="red", alpha=0.6)

p3.add_tools(HoverTool(tooltips=[
    ("Процессор", "@cpuName"),
    ("Производитель", "@manufacturer"),
    ("Ядра", "@cores")
]))

p4 = figure(title="Зависимость singleScore от базовой частоты", width=600, height=400,
            x_axis_label="Базовая частота (Гц)", y_axis_label="singleScore")
p4.scatter('baseClock', 'singleScore', source=source, size=8, color="purple", alpha=0.6)

p4.add_tools(HoverTool(tooltips=[
    ("Процессор", "@cpuName"),
    ("Производитель", "@manufacturer"),
    ("Турбо частота", "@turboClock")
]))

p5 = figure(title="Зависимость multiScore от турбо-частоты", width=600, height=400,
            x_axis_label="Турбо частота (Гц)", y_axis_label="multiScore")
p5.scatter('turboClock', 'multiScore', source=source, size=8, color="orange", alpha=0.6)

p5.add_tools(HoverTool(tooltips=[
    ("Процессор", "@cpuName"),
    ("Производитель", "@manufacturer"),
    ("Базовая частота", "@baseClock")
]))

p6 = figure(title="Зависимость singleScore от типа процессора", width=600, height=400,
            x_axis_label="Тип процессора", y_axis_label="singleScore",
            x_range=df['type'].unique()) # Уникальные значения для оси X
p6.scatter('type', 'singleScore', source=source, size=8, color=factor_cmap('type', palette=Spectral6, factors=df['type'].unique()), alpha=0.6)

p6.add_tools(HoverTool(tooltips=[
    ("Процессор", "@cpuName"),
    ("Производитель", "@manufacturer"),
    ("Тип", "@type")
]))

```

Рисунок 5 - Создание интерактивных графиков с использованием Bokeh

Графики можно увидеть на рисунках (рис. 7,8,9,10,11,12).

3.6 Компоновка и отображение графиков

Все графики объединяются в сетку 2x3 с помощью функции `gridplot` и отображаются (рис.6).

```
grid = gridplot([[p1, p2, p3], [p4, p5, p6]])  
  
show(grid)
```

Рисунок 6 - Компоновка и отображение графиков в сетке с использованием Bokeh

4.Результат

4.1. Зависимость singleScore от multiScore

Первый график (рис.7) показывает зависимость между результатами тестов `singleScore` и `multiScore`. Это позволяет оценить, как производительность процессора в однопоточных задачах соотносится с производительностью в многопоточных задачах.

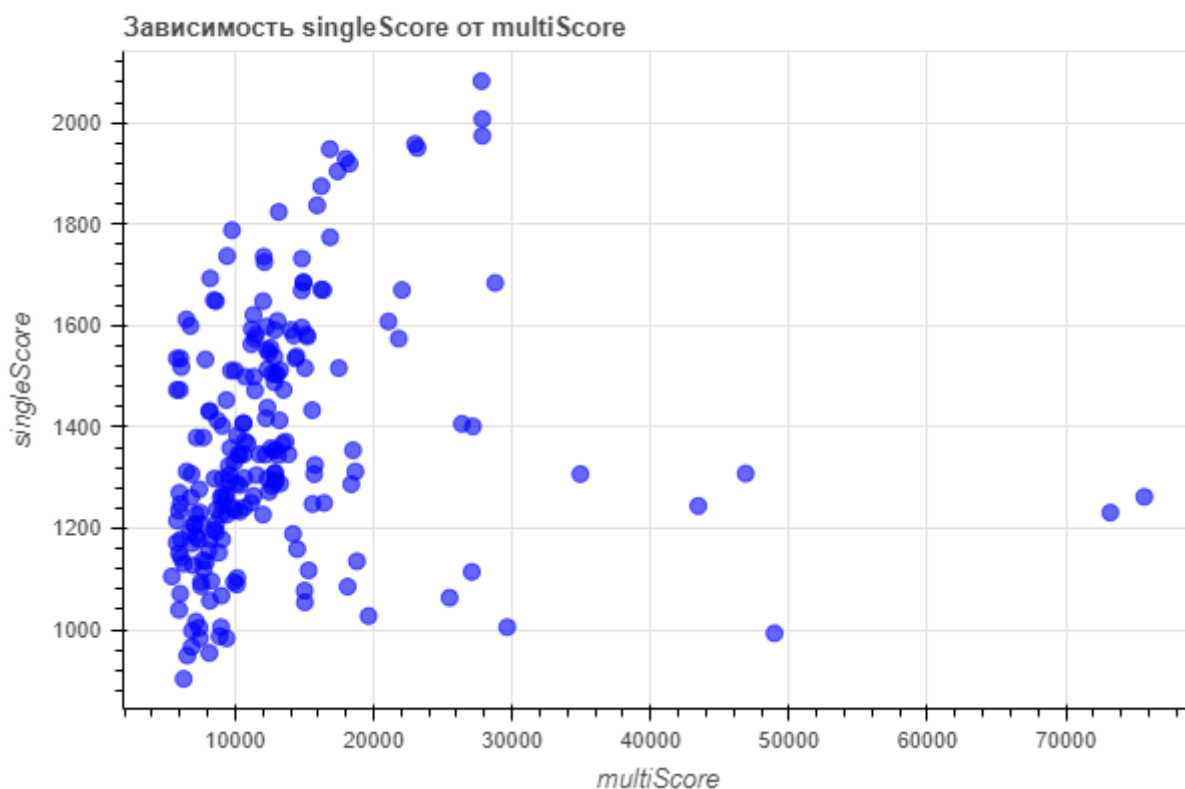


Рисунок 7- Зависимость singleScore от multiScore

4.2. Зависимость singleScore от количества ядер

Второй график (рис.8) визуализирует зависимость производительности процессора (`singleScore`) от количества ядер. Это помогает понять, как

увеличение количества ядер влияет на производительность в однопоточных задачах.

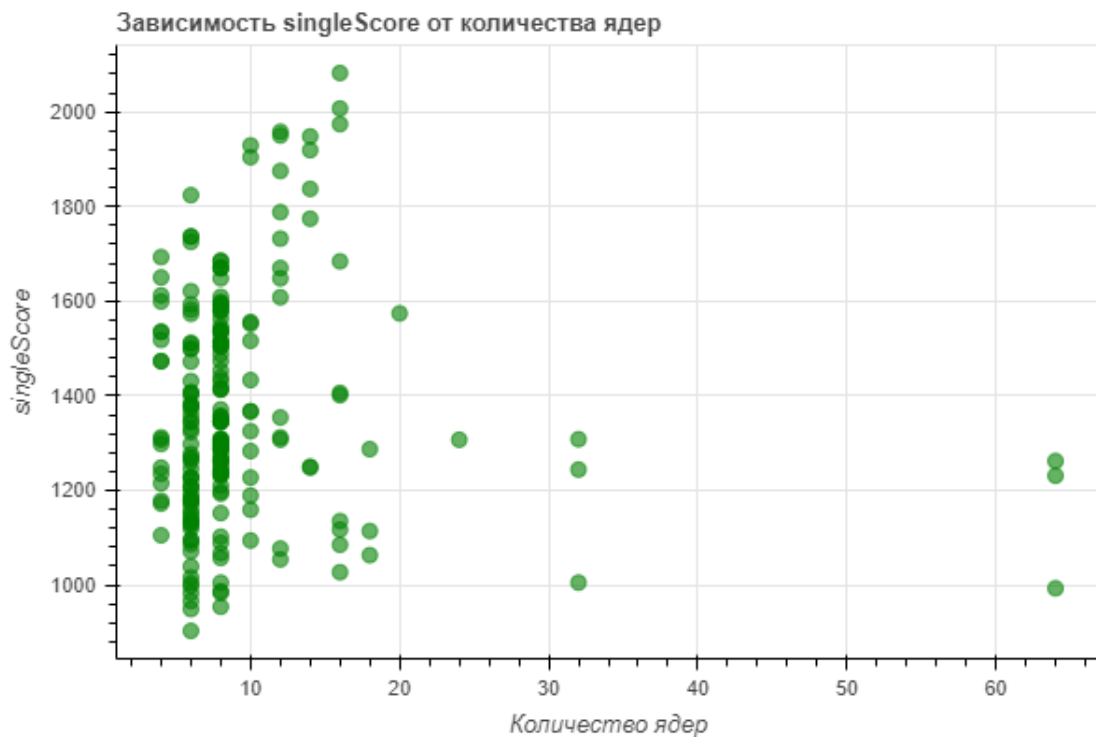


Рисунок 8 – Зависимость singleScore от количества ядер

4.3. Зависимость multiScore от количества потоков

Третий график (рис.9) показывает зависимость производительности процессора (multiScore) от количества потоков. Это позволяет оценить, как многопоточность влияет на производительность в многопоточных задачах.



4.4. Зависимость singleScore от базовой частоты

Четвертый график (рис.10) визуализирует зависимость производительности процессора (singleScore) от базовой частоты. Это помогает понять, как частота процессора влияет на его производительность.

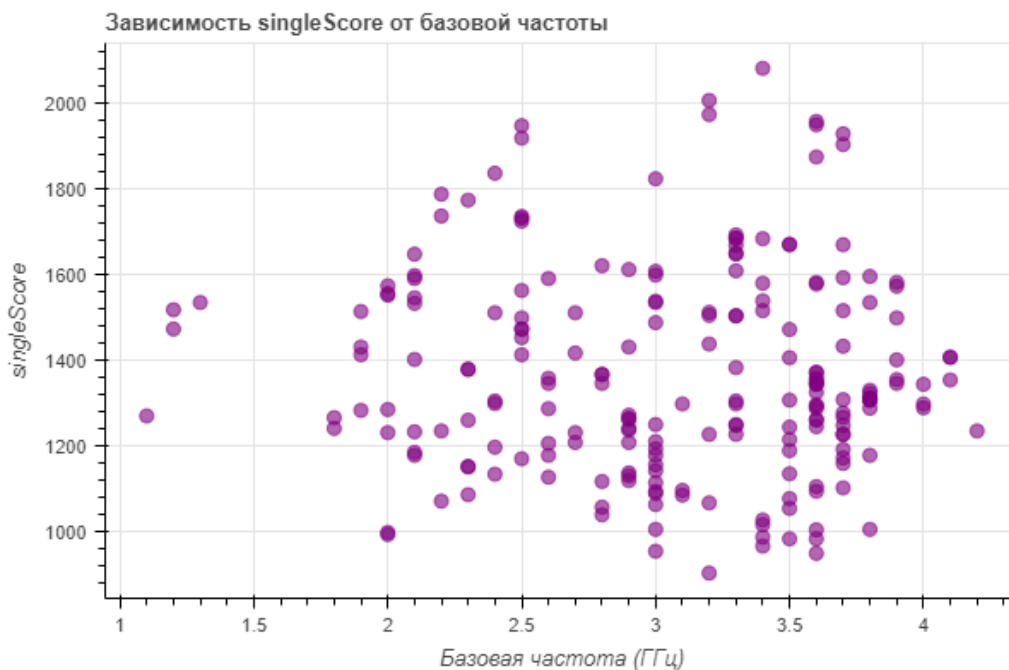


Рисунок 10 – Зависимость singleScore от базовой частоты

4.5. Зависимость multiScore от турбо-частоты

Пятый график (рис.11) показывает зависимость производительности процессора (multiScore) от турбо-частоты. Это позволяет оценить, как увеличение частоты в режиме турбо-буста влияет на производительность.

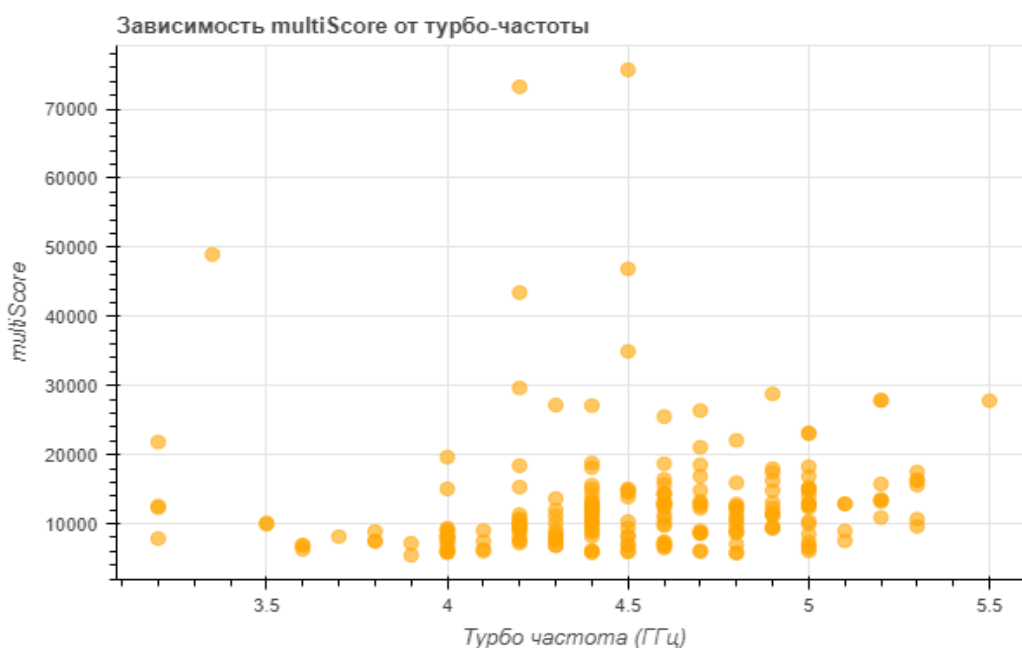


Рисунок 11 – Зависимость multiScore от турбо-частоты

4.6. Зависимость singleScore от типа процессора

Шестой график (рис.12) визуализирует зависимость производительности процессора (singleScore) от его типа (например, Desktop или Laptop). Это помогает понять, как тип процессора влияет на его производительность.

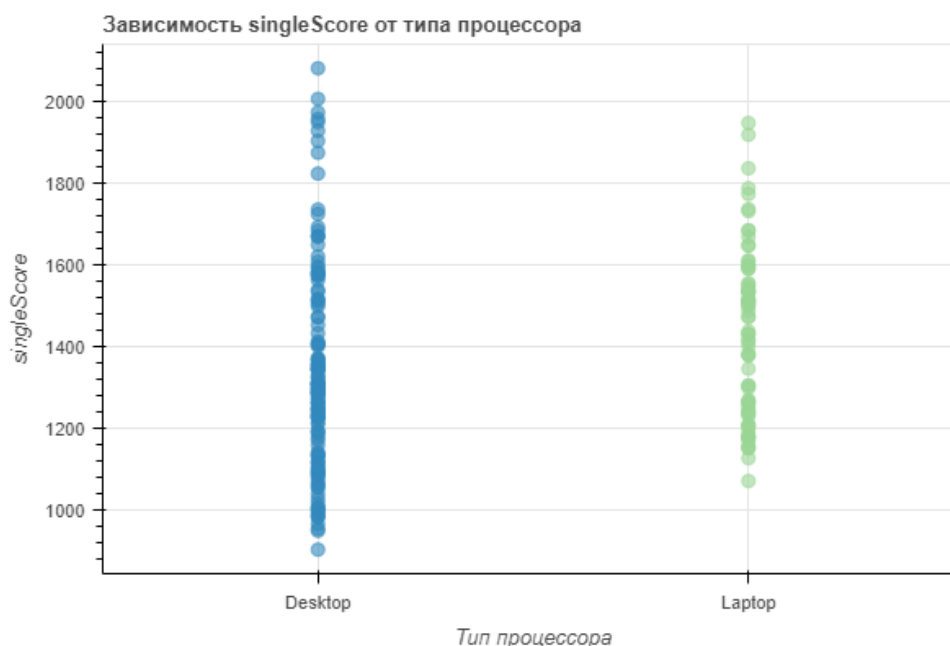


Рисунок 12 - Зависимость singleScore от типа процессора

5. Вывод

В ходе работы была успешно реализована визуализация данных о производительности процессоров с использованием библиотеки Bokeh. Созданные интерактивные графики позволили наглядно проанализировать зависимости между ключевыми характеристиками процессоров, такими как количество ядер, потоков, базовая и турбо-частота, а также результаты тестов производительности (singleScore и multiScore).

Библиографический список

1. McKinney W.. Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython. O'Reilly Media, 2017. 548 с.
2. VanderPlas J. Python Data Science Handbook: Essential Tools for Working with Data. O'Reilly Media, 2016. 541 с.
3. Grinberg M.. Flask Web Development: Developing Web Applications with Python. O'Reilly Media, 2018. 316 с.
4. Khan. A.S. Engineering a Scalable and Interactive Web Tool for JRO Data Exploration Using Python's Bokeh Library. University of Illinois at Urbana-Champaign, 2021. 120 с.
5. Chai C. et al. Interactive Visualization of Complex Seismic Data and Models Using Bokeh. Seismological Research Letters, 2018. T. 89. №. 2A. С. 668-676.