Assignment 03 – Profiling and Numba

Konstantinos Georgiou

# General Info

* The code and the results for the assignment 3 is in this GitHub repo: <https://github.com/drkostas/DSE512-playground>
* The configuration I am using to run this assignment is this one: <https://github.com/drkostas/DSE512-playground/blob/master/confs/assignment3_local_tcga.yml>
* Most of the code is in the *assignment3* folder: <https://github.com/drkostas/DSE512-playground/tree/master/assignment3>
  + There are 4 different *KMeans* implementations:
    - **simple**: Th non-vectorized *Kmeans* we created in class
    - **vectorized\_jacob**: The vectorized *Kmeans* we created in class
    - **vectorized**: The (improved) vectorized *Kmeans* version that I created in the previous assignment.
    - **jitted vectorized\_jacob**: The jitted **vectorized\_jacob** implementation after modifying it a bit to make it able be jitted.
  + I didn’t jit my vectorized implementaiton because the use of **scipy.spatial.distance.cdist**, and **np.argmin** were messing with numba’s nopython mode.
  + **assignment3.py**: Loads the configuration, and runs the appropriate *KMeans* function for each subconfig (**simple**, **vectorized\_jacob**, **vectorized**) from either **kmeans.py** or **kmeans\_numba.py**
  + **kmeans.py**: It contains the KmeansRunner class which includes all the (non-numba) Kmeans implementations and the load\_dataset() function.
  + **kmeans\_numba.py**: Contains the **jitted vectorized\_jacob** implementation. It’s not in a class
* I am also importing some other custom packages I’ve made, from which the most important ones are:
  + **profileit**: cProfile ContextManager-Decorator for profiling functions or code blocks - <https://github.com/drkostas/DSE512-playground/blob/master/playground/profiling_funcs/profileit.py>
  + **timeit**: ContextManager-Decorator for timing functions or code blocks - <https://github.com/drkostas/DSE512-playground/blob/master/playground/timing_tools/timeit.py>
* Logs:
* Screenshots:

# 1. Refactoring kmeans.py

In **kmeans.py**, there is a **run**() function which calls one of: **run\_simple()**, **run\_vectorized\_jacob(),** r**un\_vectorized()** inside a **profileit** *with* statement.

I then refactored these 3 functions, to call a **\_loop()** function which calls the **\_compute\_distances()**, **\_expectation\_step()**, **\_maximization\_step()** functions to run each individual step of the algorithm. Each implementations calls different function for these steps, for example, **run\_vectorized\_jacob()** calls **\_loop\_vectorized\_jacob()** etc. If this is not clear enough, feel free to ask me and I can elaborate more.

# 2. Profile kmeans.py

Inside **kmeans.py**, the **run()** functions, calls the appropriate implementation inside a **profileit** context manager. The times are the following:

| Algorithmic Step | Kmeans Simple | Jacob’s Kmeans Vectorized | My Kmeans Vectorized |
| --- | --- | --- | --- |
| **Compute Distances** | 90.5(s) – 99.82% | 0.531(s) – 76.17% | 0.205(s) – 50.72% |
| **Expectation** | 0.00496(s) – 0.01% | 0.00545(s) – 0.78% | 0.000122(s) – 0.03% |
| **Maximaziation** | 0.162(s) - 0.18% | 0.16(s) – 22.91% | 0.198(s) – 48.99% |
| **Total** | 90.7(s) – 100% | 0.697(s) – 100% | 0.405(s) – 100% |

The total speedup of Jacob’s Kmeans Vectorized compared to Kmeans simple is:

Jac\_vec\_perc\_increase = (90.7-0.697)/90.7\*100% = 99.231%

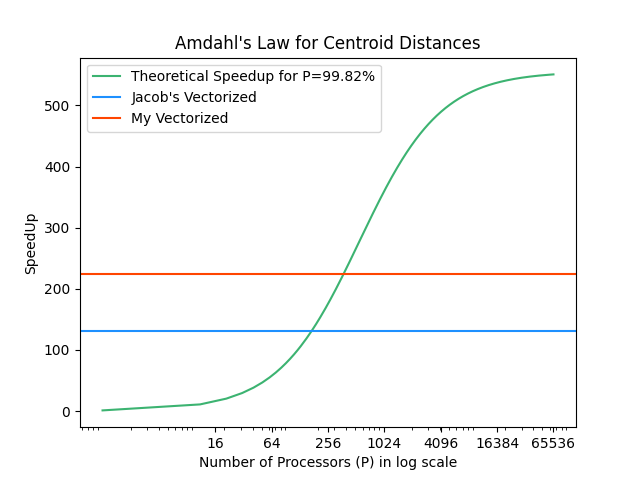
Jac\_vec\_speedup = 90.7/0.697 = 130.129

The total speedup of my Vectorized Kmeans compared to Kmeans simple is:

my\_vec\_perc\_increase = (90.7-0.405)/90.7\*100% = 99.534%

my\_vec\_speedup = 90.7/0.405 = 223.950

Plotting Amdahl’s Law with these two speedup yields the following figure:

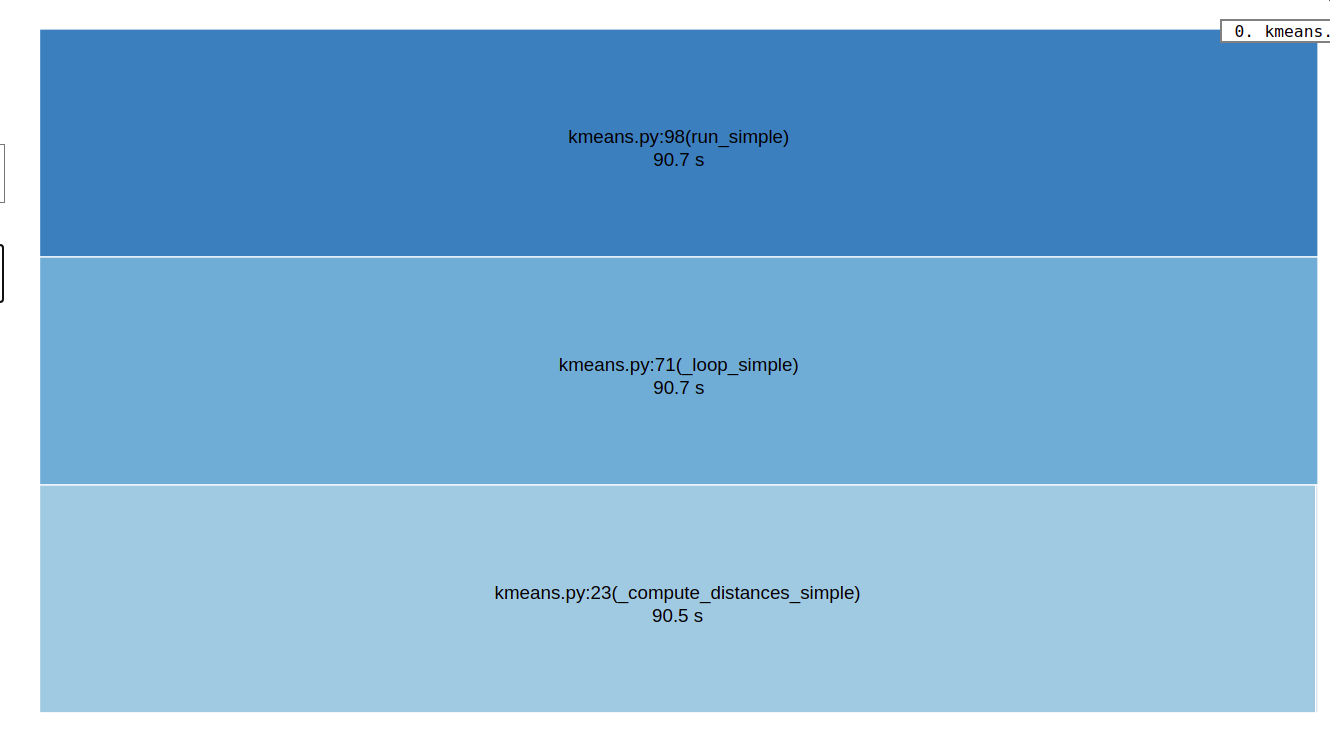


The maximum theoretical speedup that centroid distances can give if parallelized according to Amdahl’s law is **~550 times**. Jacob’s vectorized Kmeans achieved **~23.7%** of that maximum, while my vectorized Kmeans achieved **~40.7%**.

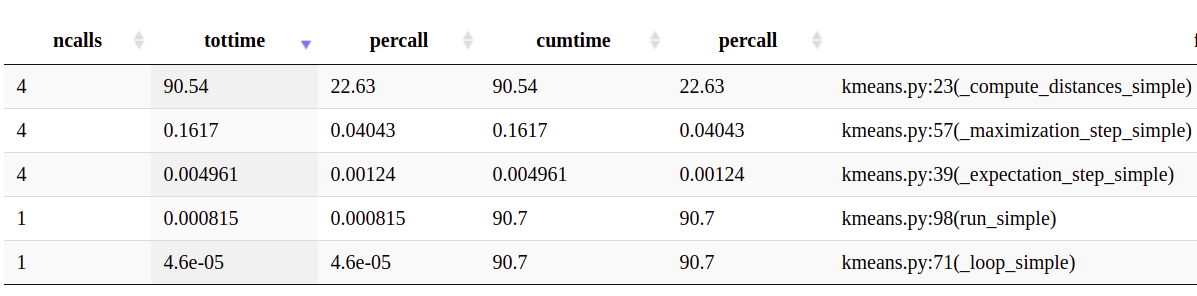
# 3. Visualize Icicle Plots

## Simple Kmeans

**Icicle Plot:**

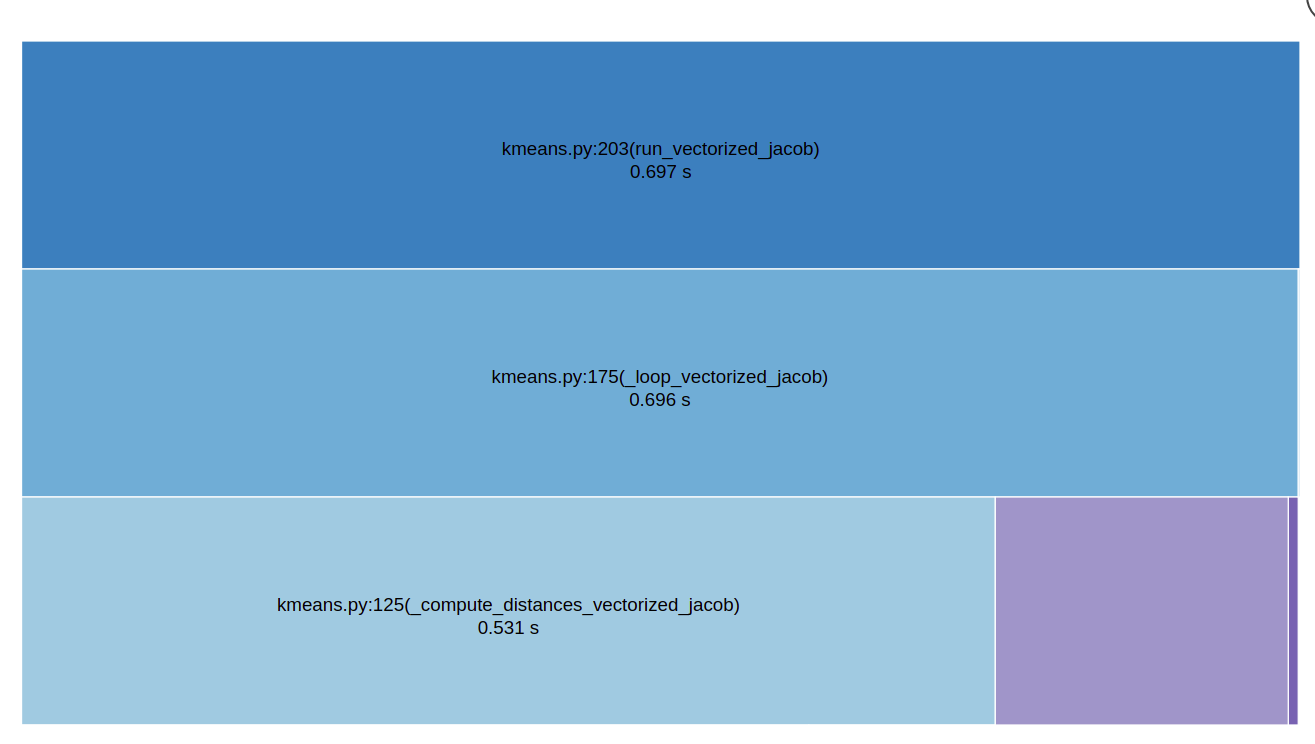


**Table with times and calls:**

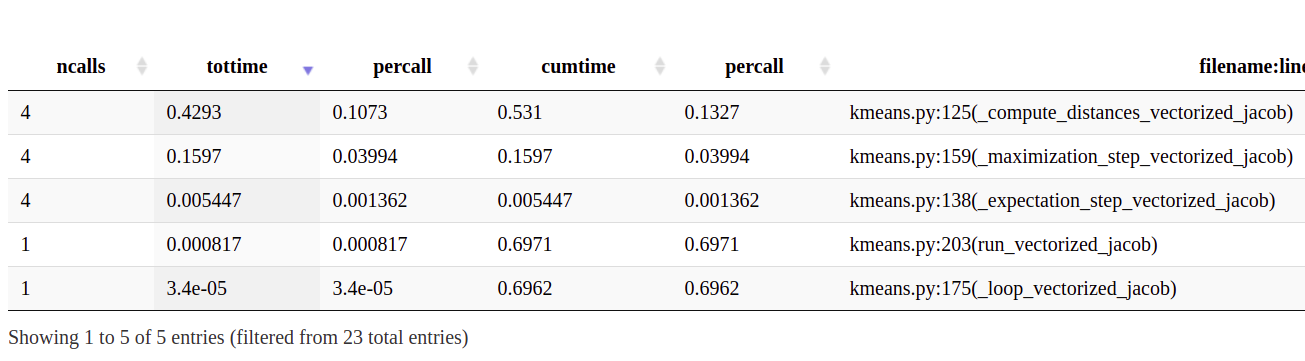


## Jacob’s Vectorized Kmeans

**Icicle Plot:**

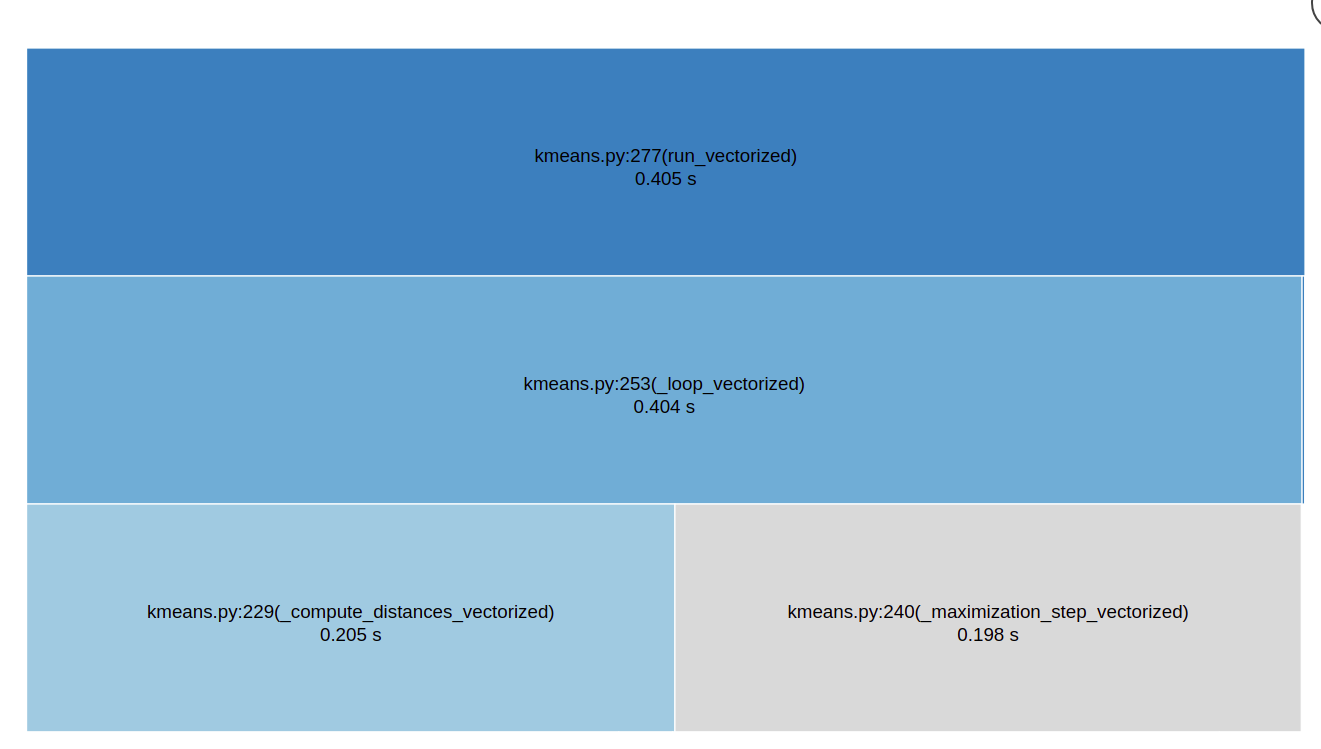


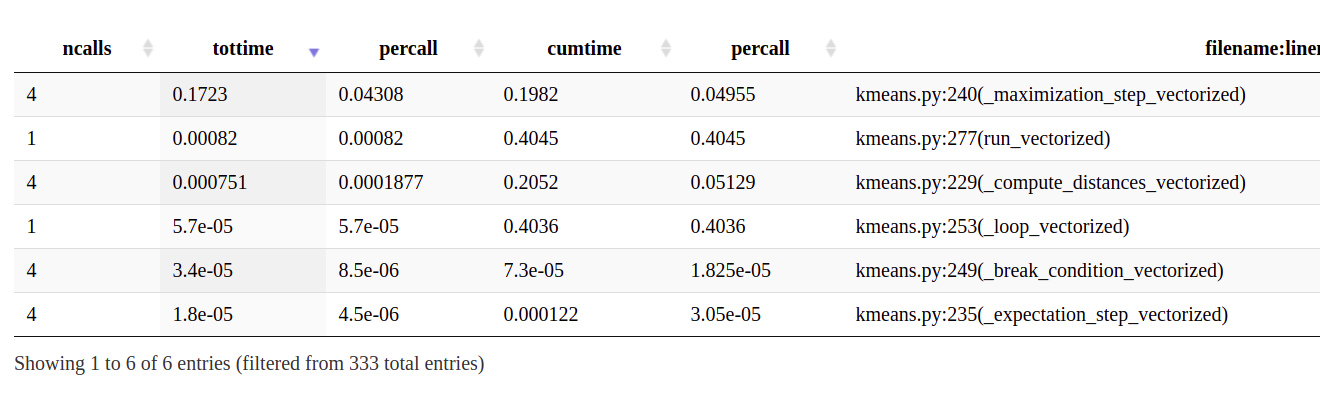
**Table with times and calls:**



## My Vectorized Kmeans

**Icicle Plot:**

****

**Table with times and calls:**

# **4. Numba**