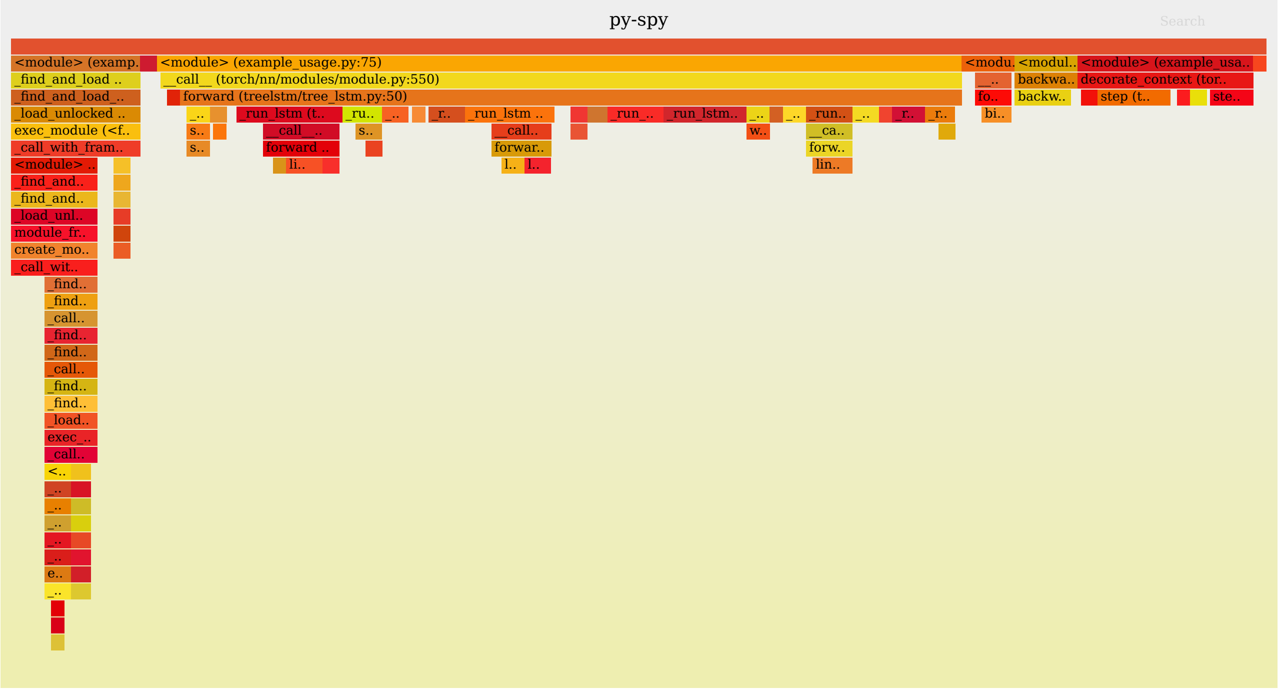
Report For Profiling Python Script

1. Py-spy (https://github.com/benfred/py-spy.git)



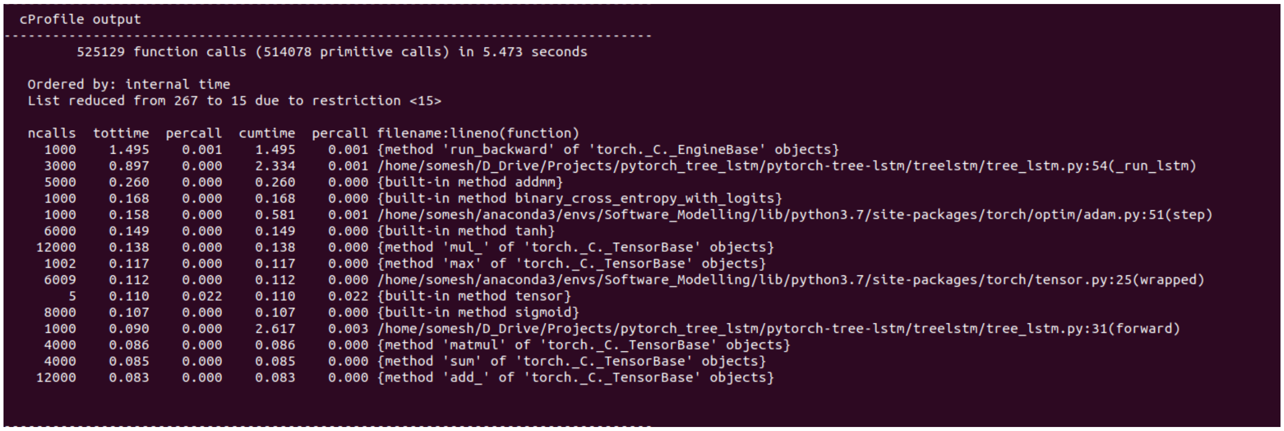
This is a simple way to generate the flame graph for the python file. It can be done using the following command. I think this is the easiest way to generate FlameGraph for any python file.

- py-spy record -o profile.svg -- python myprogram.py

2. Using torch bottleneck utility function

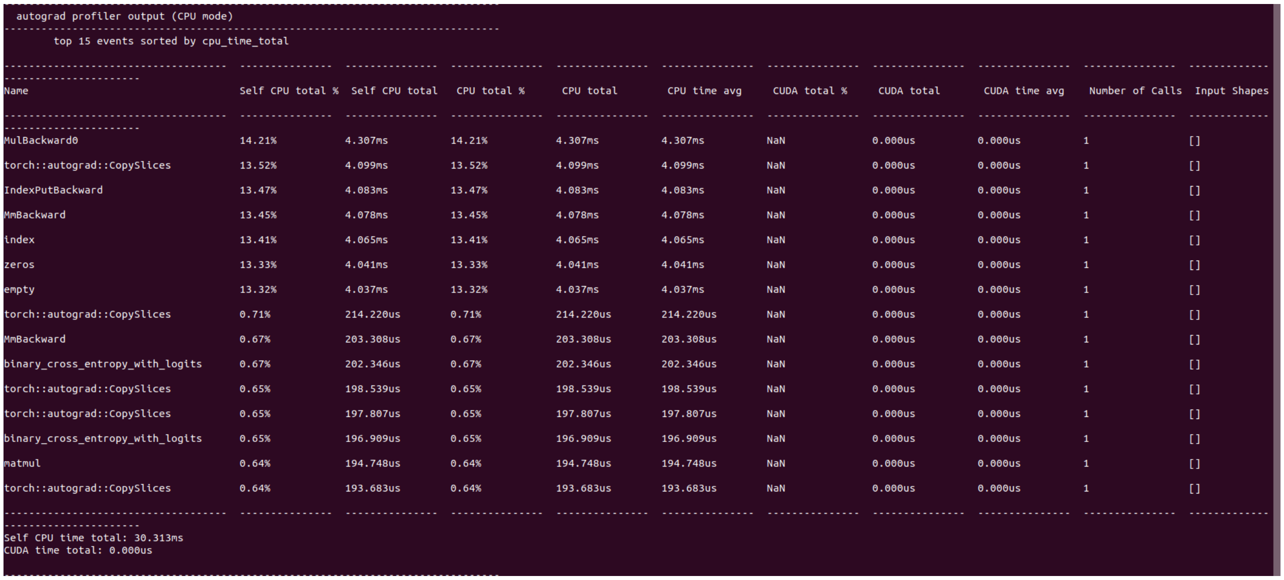
I ran this on my local system and got some data related to GPU/CPU usage which is shown in the image. I tried to get the profile data but I am not able to get the .data file so I could not generate the Flame Graph.

CProfiler_Output



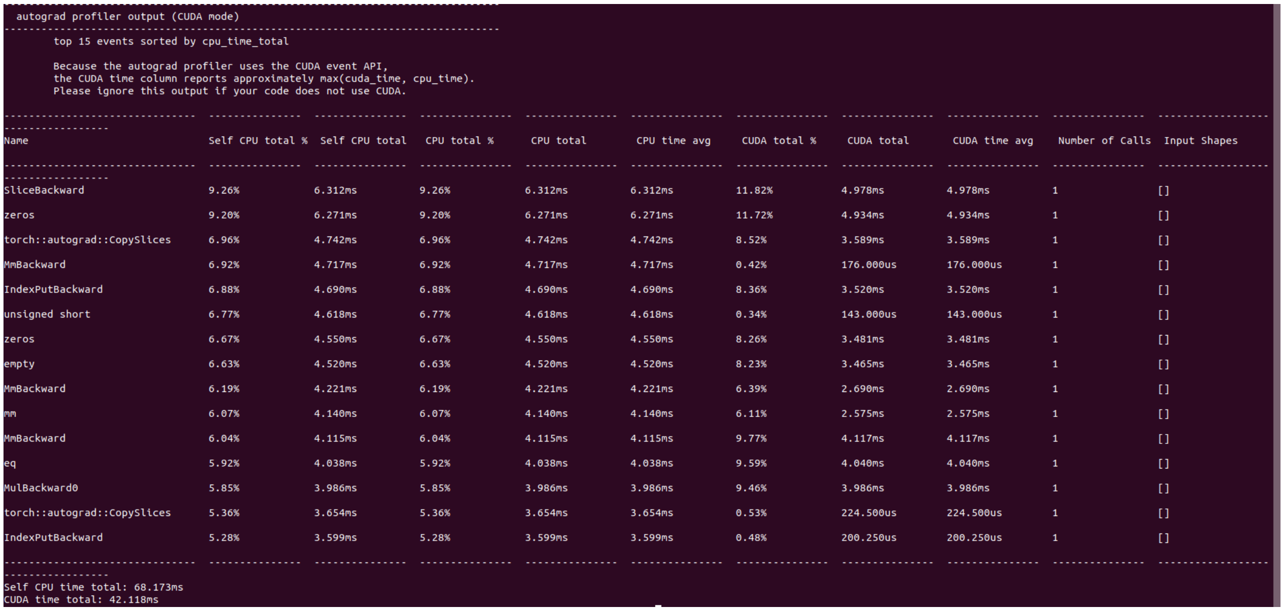
```
 cProfile output
--------------------------------------------------------------------------------
         525129 function calls (514078 primitive calls) in 5.473 seconds

   Ordered by: internal time
   List reduced from 267 to 15 due to restriction <15>

   ncalls  tottime  percall  cumtime  percall filename:lineno(function)
     1000    1.495    0.001    1.495    0.001 {method 'run_backward' of 'torch._C._EngineBase' objects}
     3000    0.897    0.000    2.334    0.001 /home/somesh/D_Drive/Projects/pytorch_tree_lstm/pytorch-tree-lstm/treelstm/tree_lstm.py:54(_run_lstm)
     5000    0.260    0.000    0.260    0.000 {built-in method addmm}
     1000    0.168    0.000    0.168    0.000 {built-in method binary_cross_entropy_with_logits}
     1000    0.158    0.000    0.581    0.001 /home/somesh/anaconda3/envs/Software_Modelling/lib/python3.7/site-packages/torch/optim/adam.py:51(step)
     6000    0.149    0.000    0.149    0.000 {built-in method tanh}
    12000    0.138    0.000    0.138    0.000 {method 'mul_' of 'torch._C._TensorBase' objects}
     1002    0.117    0.000    0.117    0.000 {method 'max' of 'torch._C._TensorBase' objects}
     6009    0.112    0.000    0.112    0.000 /home/somesh/anaconda3/envs/Software_Modelling/lib/python3.7/site-packages/torch/tensor.py:25(wrapped)
        5    0.110    0.022    0.110    0.022 {built-in method tensor}
     8000    0.107    0.000    0.107    0.000 {built-in method sigmoid}
     1000    0.090    0.000    2.617    0.003 /home/somesh/D_Drive/Projects/pytorch_tree_lstm/pytorch-tree-lstm/treelstm/tree_lstm.py:31(forward)
     4000    0.086    0.000    0.086    0.000 {method 'matmul' of 'torch._C._TensorBase' objects}
     4000    0.085    0.000    0.085    0.000 {method 'sum' of 'torch._C._TensorBase' objects}
    12000    0.083    0.000    0.083    0.000 {method 'add_' of 'torch._C._TensorBase' objects}


--------------------------------------------------------------------------------
```

AutoGrad Profiler Output CPU Mode



```
--------------------------------------------------------------------------------
  autograd profiler output (CPU mode)
--------------------------------------------------------------------------------
        top 15 events sorted by cpu_time_total

-----------------------------------  ---------------  ---------------  ---------------  ---------------  ---------------  ---------------  ---------------  ---------------  ---------------  ------------
---------------------
Name                                 Self CPU total %  Self CPU total   CPU total %      CPU total        CPU time avg     CUDA total %     CUDA total       CUDA time avg    Number of Calls  Input Shapes

-----------------------------------  ---------------  ---------------  ---------------  ---------------  ---------------  ---------------  ---------------  ---------------  ---------------  ------------
---------------------
MulBackward0                         14.21%           4.307ms          14.21%           4.307ms          4.307ms          NaN              0.000us          0.000us          1                []

torch::autograd::CopySlices          13.52%           4.099ms          13.52%           4.099ms          4.099ms          NaN              0.000us          0.000us          1                []

IndexPutBackward                     13.47%           4.083ms          13.47%           4.083ms          4.083ms          NaN              0.000us          0.000us          1                []

MmBackward                           13.45%           4.078ms          13.45%           4.078ms          4.078ms          NaN              0.000us          0.000us          1                []

index                                13.41%           4.065ms          13.41%           4.065ms          4.065ms          NaN              0.000us          0.000us          1                []

zeros                                13.33%           4.041ms          13.33%           4.041ms          4.041ms          NaN              0.000us          0.000us          1                []

empty                                13.32%           4.037ms          13.32%           4.037ms          4.037ms          NaN              0.000us          0.000us          1                []

torch::autograd::CopySlices          0.71%            214.220us        0.71%            214.220us        214.220us        NaN              0.000us          0.000us          1                []

MmBackward                           0.67%            203.308us        0.67%            203.308us        203.308us        NaN              0.000us          0.000us          1                []

binary_cross_entropy_with_logits     0.67%            202.346us        0.67%            202.346us        202.346us        NaN              0.000us          0.000us          1                []

torch::autograd::CopySlices          0.65%            198.539us        0.65%            198.539us        198.539us        NaN              0.000us          0.000us          1                []

torch::autograd::CopySlices          0.65%            197.807us        0.65%            197.807us        197.807us        NaN              0.000us          0.000us          1                []

binary_cross_entropy_with_logits     0.65%            196.909us        0.65%            196.909us        196.909us        NaN              0.000us          0.000us          1                []

matmul                               0.64%            194.748us        0.64%            194.748us        194.748us        NaN              0.000us          0.000us          1                []

torch::autograd::CopySlices          0.64%            193.683us        0.64%            193.683us        193.683us        NaN              0.000us          0.000us          1                []

-----------------------------------  ---------------  ---------------  ---------------  ---------------  ---------------  ---------------  ---------------  ---------------  ---------------  ------------
---------------------
Self CPU time total: 30.313ms
CUDA time total: 0.000us

--------------------------------------------------------------------------------
```

AutoGrad Profiler Output GPU Mode



```
--------------------------------------------------------------------------------
  autograd profiler output (CUDA mode)
--------------------------------------------------------------------------------
        top 15 events sorted by cpu_time_total

        Because the autograd profiler uses the CUDA event API,
        the CUDA time column reports approximately max(cuda_time, cpu_time).
        Please ignore this output if your code does not use CUDA.

-------------------------------  ---------------  ---------------  ---------------  ---------------  ---------------  ---------------  ---------------  ---------------  ---------------  -----------------
-----------------
Name                             Self CPU total %  Self CPU total   CPU total %      CPU total        CPU time avg     CUDA total %     CUDA total       CUDA time avg    Number of Calls  Input Shapes

-------------------------------  ---------------  ---------------  ---------------  ---------------  ---------------  ---------------  ---------------  ---------------  ---------------  -----------------
-----------------
SliceBackward                    9.26%            6.312ms          9.26%            6.312ms          6.312ms          11.82%           4.978ms          4.978ms          1                []

zeros                            9.20%            6.271ms          9.20%            6.271ms          6.271ms          11.72%           4.934ms          4.934ms          1                []

torch::autograd::CopySlices      6.96%            4.742ms          6.96%            4.742ms          4.742ms          8.52%            3.589ms          3.589ms          1                []

MmBackward                       6.92%            4.717ms          6.92%            4.717ms          4.717ms          0.42%            176.000us        176.000us        1                []

IndexPutBackward                 6.88%            4.690ms          6.88%            4.690ms          4.690ms          8.36%            3.520ms          3.520ms          1                []

unsigned short                   6.77%            4.618ms          6.77%            4.618ms          4.618ms          0.34%            143.000us        143.000us        1                []

zeros                            6.67%            4.550ms          6.67%            4.550ms          4.550ms          8.26%            3.481ms          3.481ms          1                []

empty                            6.63%            4.520ms          6.63%            4.520ms          4.520ms          8.23%            3.465ms          3.465ms          1                []

MmBackward                       6.19%            4.221ms          6.19%            4.221ms          4.221ms          6.39%            2.690ms          2.690ms          1                []

mm                               6.07%            4.140ms          6.07%            4.140ms          4.140ms          6.11%            2.575ms          2.575ms          1                []

MmBackward                       6.04%            4.115ms          6.04%            4.115ms          4.115ms          9.77%            4.117ms          4.117ms          1                []

eq                               5.92%            4.038ms          5.92%            4.038ms          4.038ms          9.59%            4.040ms          4.040ms          1                []

MulBackward0                     5.85%            3.986ms          5.85%            3.986ms          3.986ms          9.46%            3.986ms          3.986ms          1                []

torch::autograd::CopySlices      5.36%            3.654ms          5.36%            3.654ms          3.654ms          0.53%            224.500us        224.500us        1                []

IndexPutBackward                 5.28%            3.599ms          5.28%            3.599ms          3.599ms          0.48%            200.250us        200.250us        1                []

-------------------------------  ---------------  ---------------  ---------------  ---------------  ---------------  ---------------  ---------------  ---------------  ---------------  -----------------
-----------------
Self CPU time total: 68.173ms
CUDA time total: 42.118ms
```

I will try to find the issue with generating .data file to generate the flame graph for this data.

3. Pyflame FlameGraph

I tried this library pyflame (https://github.com/uber-archive/pyflame) for generating the FlameGraph (https://github.com/brendangregg/FlameGraph) for the python script but the .data file is not generated. I get the Error "Failed to locate libpython within timeout period". Further analysis into the error I found this post (https://github.com/uber-archive/pyflame/issues/120) which says that there are some issue with python 3.7 version for generation of flamegraph. I think this library can be used for generating the FlameGraph by using python 3.4 version. As for now I have python 3.7.7 on my system so I will setup a venv for this and try to generate flame graph using this library.

I have update all the output mentioned here in the below github link for your reference.

- https://github.com/somesh636/Software_Modelling_AI (https://github.com/somesh636/Software_Modelling_AI)

I have successfully run the example_usage.py file on the google colab just to check if it can be done. The file "Profiling_lstm.ipynb" can be directly run on to google colab which is a copy of example_usage.py file.