Performance Measurement of the C++ and python image processing pipeline

Atmadeep Arya; Candidate System Software Engineer @ Vimaan.ai

August 19, 2024

Performance Measurement of the C++ Pipeline

1. **Latency of the pipeline**: 0.25 Seconds/Frame

2. **Throughput of the pipeline**: 4.125 Frames/Second

3. **Processing function's metrics**

   (a) Throughput: 4.12 Frames/Second

   (b) Latency: 0.24 Seconds/Frame

   (c) This is done in order to measure the performance of the core processing function.

Memory leaks and usage

1. Currently, the memory leaked by `gtk` applications used by `OpenCV` in `displayImage()` function will cause some memory leaks. This is checked using `AddressSanitizer`

   - #1 0x7fad8f964e98 in `g_malloc`: This indicates that the `g_malloc` function from `libglib-2.0.so.0` called `malloc`. The actual allocation is happening inside the glib library, which is widely used in GNOME applications and by GTK.

```
==323047==ERROR: LeakSanitizer: detected memory leaks

Direct leak of 16384 byte(s) in 1 object(s) allocated from:
    #0 0x7fcb1bda1808 in __interceptor_malloc ../../../../src/
      libsanitizer/asan/asan_malloc_linux.cc:144
    #1 0x7fcb18bece98 in g_malloc (/lib/x86_64-linux-gnu/libglib
      -2.0.so.0+0x57e98)

Direct leak of 6168 byte(s) in 3 object(s) allocated from:
    #0 0x7fcb1bda3787 in operator new[](unsigned long)
      ../../../../src/libsanitizer/asan/asan_new_delete.cc:107
   profiler #1 0x7fcb1282b66f (/lib/x86_64-linux-gnu/libtbb.so
      .2+0x2766f)

SUMMARY: AddressSanitizer: 22552 byte(s) leaked in 4 allocation(s)
  .
```

2. Checking the memory used by various functions, we've used `heaptrack` for memory usage profiling.

- **Total memory used**: 36.6 MB after 10.77s of running the pipeline
- **captureImage()** → 7.1 MB (5.69% contribution)
- **processImage()** → 2.1 MB (19.4% contribution)
- **displayImage()** → 9.7 MB (26.1% contribution)

Python performance measurement

Time and memory taken:

1. Memory taken : Proportional set size = 58.57 MB after 11 seconds of run time.

2. Latency and throughput are not measured for python application as the entire binary is packaged as single function.

3. Call graph for the python pipeline is attached and can be visualized using snakeviz.