

# GstCUDA: A GStreamer-CUDA interface solution

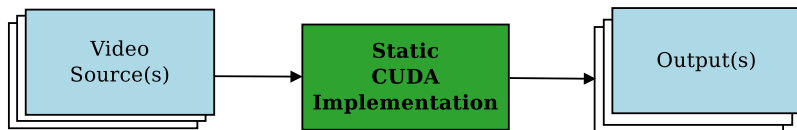
José Jiménez-Chavarría

RidgeRun Engineering  
GStreamer Conference 2018

October 25, 2018



# Traditional Solutions

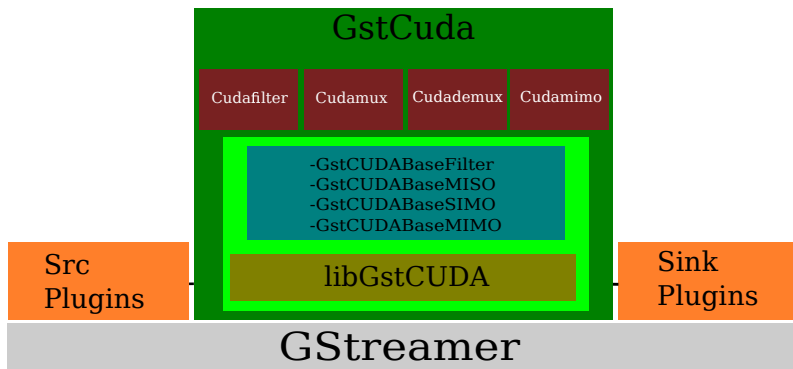


## Disadvantages

- Doesn't scale properly when adding/removing inputs/outputs
- No complexity abstraction
- Doesn't reach peak performance

# GstCUDA

GstCuda offers a framework to develop custom GStreamer/CUDA elements:



# Features

## Key features

- Framework supports GStreamer elements that can execute any CUDA algorithm
- Zero memory copy (NVMM direct mapping & Unified memory allocator) on Jetson
- High performance from the get go using prototyping elements: **cudafilter**, **cuda mux**, **cuda demux**, **cuda mimo**.

# Example 1

## Gray-scale filter using **cudafilter** on TX1 (NVMM) memory



### 4K 60fps capture to display (in-place=true)

#### Example pipeline

```

gst-launch-1.0 nvcamerasrc queue-size=10 sensor-id=0 fpsRange='60 60' ! "video/x-raw(memory:NVMM),width=3840,height=2160,format=I420,framerate=60/1" ! nvvidconv ! "video/x-raw(memory:NVMM),width=3840,height=2160,format=I420,framerate=60/1" ! cudafilter in-place=true location=./gray-scale-filter.so ! perf print-arm-load=true ! nvoverlaysink --gst-debug=0
  
```

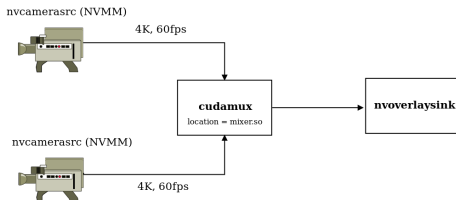
#### Performance stats

```

GST-PERF INFO --> Timestamp: 0:03:56.819087040; Bps: 776; fps: 60.0; CPU: 16;
GST-PERF INFO --> Timestamp: 0:03:57.825121037; Bps: 771; fps: 59.64; CPU: 13;
GST-PERF INFO --> Timestamp: 0:03:58.837141233; Bps: 766; fps: 59.28; CPU: 14;
GST-PERF INFO --> Timestamp: 0:03:59.849401695; Bps: 766; fps: 59.28; CPU: 14;
GST-PERF INFO --> Timestamp: 0:04:00.859525079; Bps: 768; fps: 59.40; CPU: 13;
GST-PERF INFO --> Timestamp: 0:04:01.871618207; Bps: 766; fps: 59.28; CPU: 15;
GST-PERF INFO --> Timestamp: 0:04:02.883459413; Bps: 767; fps: 59.34; CPU: 15;
GST-PERF INFO --> Timestamp: 0:04:03.894646509; Bps: 767; fps: 59.34; CPU: 14;
GST-PERF INFO --> Timestamp: 0:04:04.906216370; Bps: 767; fps: 59.34; CPU: 14;
GST-PERF INFO --> Timestamp: 0:04:05.917121183; Bps: 768; fps: 59.40; CPU: 14;
  
```

# Example 2

## Video mixing using **cudamux** on TX1 (NVMM) memory



2x 4K 60fps camera streams camera streams mixed to display (in-place=true)

Example pipeline

```
gst-launch-1.0 cudamux name=cuda in-place=true location=./mixer.so nvcamerasrc queue-size=10 sensor-id=0 fpsRange='60 60' ! "video/x-raw(memory:NVMM),width=3840,height=2160,format=I420,framerate=60/1" ! tee name=t t.src_0 ! nvvidconv ! "video/x-raw(memory:NVMM),width=3840,height=2160,format=I420,framerate=60/1" ! queue ! cuda.sink_0 t.src_1 ! nvvidconv flip-method=2 ! "video/x-raw(memory:NVMM),width=3840,height=2160,format=I420,framerate=60/1" ! queue ! cuda.sink_1 cuda. ! perf print-arm-load=true ! nvoverlaysink --gst-debug=0
```

Performance stats

```
GST-PERF INFO --> Timestamp: 1:09:53.580481344; Bps: 776; fps: 60.0; CPU: 16;
GST-PERF INFO --> Timestamp: 1:09:54.596964159; Bps: 763; fps: 60.3; CPU: 18;
GST-PERF INFO --> Timestamp: 1:09:55.612936985; Bps: 764; fps: 60.9; CPU: 17;
GST-PERF INFO --> Timestamp: 1:09:56.613082828; Bps: 776; fps: 60.0; CPU: 20;
GST-PERF INFO --> Timestamp: 1:09:57.622503818; Bps: 769; fps: 59.46; CPU: 18;
GST-PERF INFO --> Timestamp: 1:09:58.634003692; Bps: 767; fps: 59.34; CPU: 18;
GST-PERF INFO --> Timestamp: 1:09:59.646157364; Bps: 766; fps: 59.28; CPU: 16;
GST-PERF INFO --> Timestamp: 1:10:00.657026108; Bps: 768; fps: 59.40; CPU: 15;
GST-PERF INFO --> Timestamp: 1:10:01.667696004; Bps: 768; fps: 59.40; CPU: 18;
GST-PERF INFO --> Timestamp: 1:10:02.680801291; Bps: 766; fps: 59.23; CPU: 17;
```

# References

More information in:

[developer.ridgerun.com/  
wiki/index.php?title=GstCUDA](http://developer.ridgerun.com/wiki/index.php?title=GstCUDA)

Thanks!

