COLLABORA

From Streams to Insights Advancing GstAnalytics

Daniel Morin

Collabora

October 23, 2025

Open First

Analytics pipeline

What is an analytics pipeline.

Analytics pipeline

COLLABORA

Presentation Outline

Gst-Analytics advancements this year

Python Bindings

Tensor-id Registry

Tensor negotiation

Gst-Python-ML

Gst-Analytics advancements this year

- Enhanced metadata and machine learning support

Gst-Analytics advancements this year

- Enhanced metadata and machine learning support
- Pythonic bindings

Gst-Analytics advancements this year

- Enhanced metadata and machine learning support
- Pythonic bindings
- Adoption of Tensor-Id-Registry

Gst-Analytics advancements this year

- Enhanced metadata and machine learning support
- Pythonic bindings
- Adoption of Tensor-Id-Registry
- Tensor negotiation and tensordecodebin

Gst-Analytics advancements this year

- Enhanced metadata and machine learning support
- Pythonic bindings
- Adoption of Tensor-Id-Registry
- Tensor negotiation and tensordecodebin
- Gst-Python-ML

Enhancement to Metadata and ML supports

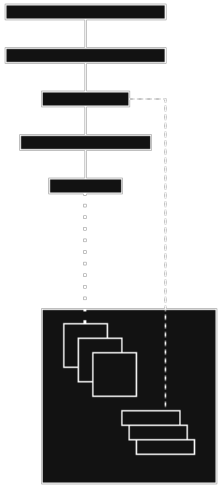
- GstTensor extracted from GstTensorMeta

Enhancement to Metadata and ML supports

- GstTensor extracted from GstTensorMeta
- GstAnalyticsBatchMeta

Enhancement to Metadata and ML supports

- GstTensor extracted from GstTensorMeta
- GstAnalyticsBatchMeta
- GstAnalyticsTensorMtd



Enhancement to Metadata and ML supports

- GstTensor extracted from GstTensorMeta
- GstAnalyticsBatchMeta
- GstAnalyticsTensorMtd
- GstAnalyticsObjectDetectionMtd now supports rotation

Enhancement to Metadata and ML supports

- GstTensor extracted from GstTensorMeta
- GstAnalyticsBatchMeta
- GstAnalyticsTensorMtd
- GstAnalyticsObjectDetectionMtd now supports rotation
- Added validation for tensor retrieval from GstTensorMeta

Enhancement to Metadata and ML supports

- GstTensor extracted from GstTensorMeta
- GstAnalyticsBatchMeta
- GstAnalyticsTensorMtd
- GstAnalyticsObjectDetectionMtd now supports rotation
- Added validation for tensor retrieval from GstTensorMeta
- New N-to-N relation type between Mtd. Used in segmentation Between ClsMd and Segmentation

COLLABORA

Presentation Outline

Gst-Analytics advancements this year

Python Bindings

Tensor-id Registry

Tensor negotiation

Gst-Python-ML

Pythonic bindings

Pythonic bindings

- Iterator on GstAnalytics.RelationMeta

```
# Extract from test_analytics.py

buf = Gst.Buffer()
rmeta = Gst-Analytics.buffer_add_analytics_relation_meta(buf)

rmeta.add_od_mtd(GLib.quark_from_string("od"), 1, 1, 2, 2, 0.1)

#...

mtds_from_iter = list(rmeta)

# or
cls_mtds = list(rmeta.iter_on_type(GstAnalytics.ClsMtd))

# List comprehension
cls_mtds2 = [mtd for mtd in mtds_from_iter if type(mtd) == GstAnalytics.ClsMtd]
```

COLLABORA

Presentation Outline

Gst-Analytics advancements this year

Python Bindings

Tensor-id Registry

Tensor negotiation

Gst-Python-ML

Tensor-id Registry

Tensor-Id registry is a git repository where tensor layouts are described and assigned a unique ID.

- Repository where everyone can use and contribute by registering tensor encoding
- Description are code agnostic

https://github.com/collabora/tensor-id-registry

COLLABORA

Presentation Outline

Gst-Analytics advancements this year

Python Bindings

Tensor-id Registry

Tensor negotiation

Gst-Python-ML

Tensor negotiation

- Validate compatibility between elements producing and consuming tensors

Tensor negotiation

- Validate compatibility between elements producing and consuming tensors
- Allow creation of tensordecodebin

Tensor negotiation

- Validate compatibility between elements producing and consuming tensors
- Allow creation of tensordecodebin
- Allow selection of tensordecoder based on compatibility and rank

Tensor negotiation

- Validate compatibility between elements producing and consuming tensors
- Allow creation of tensordecodebin
- Allow selection of tensordecoder based on compatibility and rank
- Potentially reduce dependency of analytics pipeline to a specific platform

Tensor negotiation

Pipeline without tensor caps

```
gst-launch-1.0 ... ! onnxinference model-file=ssd_mobilenet_v1_coco.onnx ! classifiertensordecoder ! ...
```

Tensor negotiation

Pipeline with tensor caps

```
gst-launch-1.0 ... ! onnxinference model-file=ssd_mobilenet_v1_coco.onnx ! classifiertensordecoder ! ...
```

Tensor negotiation

Not fixed classifiertensordecoder caps

```
/classifiertensordecoder.GstPad.sink:
 tensors =
   "tensorgroups\,
     classification-generic-out=(/set){
       (caps)"tensor/strided,
         tensor-id=(string)classification-generic-out,
         dims=(int)< [ 0, 1 ], [ 1, 2147483647 ] >,
         dims-order=(string)row-major,
         type=(string){ float32, uint8 };
       (caps)tensor/strided,
         tensor-id=(string)classification-generic-out,
         dims=(int)< [ 1, 2147483647 ] >,
         dims-order=(string)row-major,
         type=(string){ float32, uint8 }"
     };"
```

COLLABORA

Tensor negotiation

Not fixed ssd tensor decoder caps

```
/ssdobjectdetector.GstPad.sink:
  tensors=
    "tensorgroups,
      ssd-mobilenet-v1-variant-1-out=(/set){
        (caps)"tensor/strided,
          tensor-id=(string)generic-variant-1-out-count,
          dims=(int)< [ 0, 2147483647 ] >,
          dims-order=(string)row-major,
          type=(string)float32",
        (caps)"tensor/strided,
          ...
        (caps)"tensor/strided,
          tensor-id=(string)ssd-mobilenet-v1-variant-1-out-boxes,
          dims=(int)< [ 0, 2147483647 ], 0, 4 >,
          dims-order=(string)row-major,
          type=(string)float32",
        (caps)"tensor/strided,
          tensor-id=(string)ssd-mobilenet-v1-variant-1-out-classes,
          dims=(int)< [ 0, 2147483647 ], 0 >,
          dims-order=(string)row-major,
          type=(string)float32"
      };"
```

Tensor negotiation

onnxinference with ssd model

```
onnxinference0.GstPad:src:
  caps = video/x-raw,
  ...
  tensors=(structure)[
    tensorgroups,
      classification-generic-out=(/set){
        (caps)"tensor/strided,
          dims-order=(string)row-major,
          dims=(int)< 0, 1000 >,
          tensor-id=(string)classification-generic-out,
          type=(string)float32" };
  ]
```

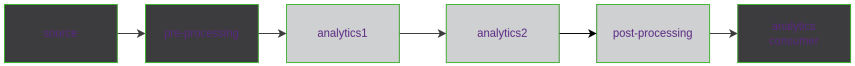
COLLABORA

Tensor negotiation

Fixed classifiertensordecoder caps

```
classifiertensordecoder0.GstPad:sink:
  caps = video/x-raw,
    width=(int)224,
    height=(int)224,
    tensors=(structure)[
      tensorgroups,
        classification-generic-out=(/set){
          (caps)"tensor/strided,
            dims-order=(string)row-major,
            dims=(int)< 0, 1000 >,
            tensor-id=(string)classification-generic-out,
            type=(string)float32" };
    ]
```

Tensor negotiation



Fixed classifiertensordecoder caps

```
gst-launch-1.0 ... \
  ! onnxinference model-file=ssd_mobilenet_v1_coco.onnx ! ...
  ! onnxinference model-file=mobilenetv2-10.onnx
  ! tensordecodebin ! ...
```

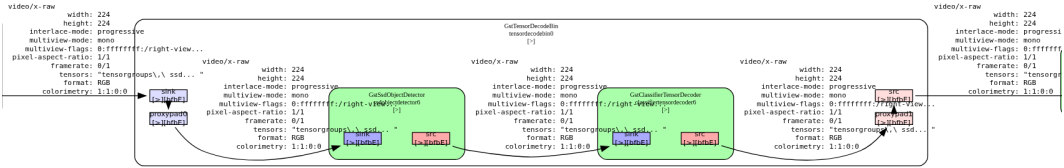
Tensor negotiation

Fixed tensordecodebin caps

```
tensordecodebin0.GstGhostPad:sink:
 caps = video/x-raw,
   format=(string)RGB, ...,
   tensors=(structure)[
    tensorgroups,
     ssd-mobilenet-v1-variant-1-out=(/set){
       (caps)"tensor/strided,
         dims-order=(string)row-major,
         dims=(int)< 0, 0, 4 >,
         tensor-id=(string)ssd-mobilenet-v1-variant-1-out-boxes,
         type=(string)float32",
       (caps)"tensor/strided,
         ...
     },
     classification-generic-out=(/set){
       (caps)"tensor/strided,
         dims-order=(string)row-major,
         dims=(int)< 0, 1000 >,
         tensor-id=(string)classification-generic-out,
         type=(string)float32" };
   ], colorimetry=(string)1:1:0:0
```

COLLABORA

Tensor negotiation



COLLABORA

Presentation Outline

Gst-Analytics advancements this year

Python Bindings

Tensor-id Registry

Tensor negotiation

Gst-Python-ML

Open First

Gst-Python-ML

Gst-python-ml based on GStreamer and provides elements implemented in Python.

Roadmap

- Rework inference elements to be multi-modal

Roadmap

- Rework inference elements to be multi-modal
- Modify onnxinference to use ModelInfo like tfliteinference

Roadmap

- Rework inference elements to be multi-modal
- Modify onnxinference to use ModelInfo like tfliteinference
- Add support for floating point video-format

Roadmap

- Rework inference elements to be multi-modal
- Modify onnxinference to use ModelInfo like tfliteinference
- Add support for floating point video-format
- Add support for specialized pre-processing like mean and range normalization

Roadmap

- Rework inference elements to be multi-modal
- Modify onnxinference to use ModelInfo like tfliteinference
- Add support for floating point video-format
- Add support for specialized pre-processing like mean and range normalization
- Improve acceleration support

Gst-Python-ML

Gst-Python-ML in action

Thanks!

Q & A