Modern tools to debug GStreamer applications

Guillaume Desmottes

guillaume.desmOTTes@collabora.com

3rd February 2018
Plan

- Tracers
- Leak tracer
- GstShark
- Dealing with logs
GstTracer

- Introduced in 1.8
- Tracing module loaded at run time
- Post-run analysis and live introspection
- Monitoring hooks
- Formatted output
Stats tracer

- $ GST_DEBUG_FILE=stats.log GST_DEBUG="GST_TRACER:7" GST_TRACERS="stats;rusage" gst-launch-1.0 videotestsrc num-buffers=10 ! x264enc ! fakesink

- $ gst-stats-1.0 stats.log

Overall Statistics:
Number of Threads: 2
Number of Elements: 3
Number of Bins: 1
Number of Pads: 4
Number of GhostPads: 0
Number of Buffers passed: 20
Number of Events sent: 12
Number of Messages sent: 53
Number of Queries sent: 45
Time: 0:00:00.070907001
Avg CPU load: 32.8%

Thread 0x259c100 Statistics:
Time: 0:00:00.041466598
Avg CPU load: 58.4%

Pad Statistics:
videotestsrc0.src: buffers 10 (live 0, dec 0, dis 0, res 0, cor 0, mar 0, hdr 0, gap 0, drop 0, dlt 0), size (min/avg/max) ......./ 236480/ ....... time 0:0
x264enc0.src: buffers 10 (live 0, dec 0, dis 0, res 0, cor 0, mar 0, hdr 0, gap 0, drop 0, dlt 0), size (min/avg/max) 4747/ 5165/ 5549, time 0:0

Element Statistics:
GstVideoTestSrcvideotestsrc0: buffers in/out 18/ 10 bytes in/out 2364000/ 2364000 first activity 0:00:00.0376282819, ev/msg/qry sent 4/ 6/ 10
Gstx264Encx264enc0: buffers in/out 18/ 10 bytes in/out 2364000/ 2364000 first activity 0:00:00.0376282819, ev/msg/qry sent 4/ 6/ 2
GstFakeSinkfakesink0: buffers in/out 18/ bytes in/out 51673/ - first activity 0:00:00.0531528232, ev/msg/qry sent 1/ 11/ 14
Latency tracer

- Measure the time it took for each buffer to travel from source to sink.

- $ GST_DEBUG="GST_TRACER:7"$ GST_TRACERS=latency gst-launch-1.0 v4l2src ! videoconvert !
x264enc tune=zerolatency ! queue ! avdec_h264 ! queue ! glimagesink
Leaks tracer

- Track refcounts of GObject and GstMiniObject
  - Only track leaks in gst code!
- Raise a warning on leaks
- Integrated in 1.10 (core)
- No false positives
- Much lighter/faster than Valgrind
Tracking leaks

- $ GST_DEBUG="GST_TRACER:7" GST_TRACERS=leaks gst-launch-1.0 v4l2src num-buffers=10 ! 
  videoconvert ! x264enc tune=zerolatency ! queue ! avdec_h264 ! queue ! glimagesink
Tracking leaks: stack trace

- Libunwind
- GST_TRACERS="leaks(stack-traces-flags=full,filters=GstGLDisplayWayland)"
Leaks tracer: extra features

• Track ref/unref operations (check-refs=true)
• List alive objects while running (SIGUSR1)
• Check points (SIGUSR2)
GstShark: tracers

- Inter latency
- Buffer rate on src pad
- Schedule time
- Queues level
- ...
- https://github.com/RidgeRun/gst-shark
GstShark: visualization

Frame rate and CPU usage

- "videorate0_src"
- "videoconvert0_src"
- "avimux0_src"
- "queue0_src"
- "avenc_h263p0_src"
- "queue1_src"
- "capsfilter0_src"
- "videofiltersrc0_src"
- CPU average
Dealing with GStreamer logs
GStreamer Debug Viewer

gst-log - GStreamer Debug Viewer

Application | Time | Category | Code | Function | Description | Message |
--- | --- | --- | --- | --- | --- | --- |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
gst-log-parser

- GStreamer logs parsing library
- High level objects to manipulate logs
- Easy filtering, mapping, etc (iterator)
- https://github.com/gdesmott/gst-log-parser/
Gst-log-parser: ts-diff

- Highlight highest timestamps gaps (per thread)
Modern tools to debug GStreamer applications

Any questions?