Performance evaluation of chromium through task modeling and critical path analysis

Majid Rezazadeh
Vahid Azhari

Dec 6, 2018

Polytechnique Montréal
Laboratoire DORSAL
Agenda

- Motivation
- An overview of task scheduling in Chromium
- Chromium tracer
- Multi-level analysis
- Detection of root cause in more details
- Finding performance bugs and problems
- Trace Compass views
- Conclusion
- Future work
Motivation

• Identifying performance degradation and bottlenecks of chromium as a complex multi-threaded program
• Extracting all the important tracing data from a complex shared memory application
• Multi-level analysis of chromium in order to obtain a comprehensive insight into performance problems and issues
• Critical path analysis as an efficient approach to understand the Chromium behavior and enhance its responsiveness
An overview of task scheduling in Chromium

Task

• A task is a unit of work to be executed
• Each task as an object inside the task scheduler has:
  • a closure
  • traits
  • post timestamp
An overview of task scheduling in Chromium

Sequence

- The tasks in a sequence are executed in the order they were added to the sequence.

A B C D

‘A’ is the next task to execute in this sequence

Parallel tasks

E

F
An overview of task scheduling in Chromium

Priority Queue (and SequenceSortKey)

- Sequences which are waiting to be run live in a priority queue
- The SequenceSortKey is used to change the priority of the sequence after being posted

Priority Queue

SequenceSortKey
An overview of task scheduling in Chromium

Scheduling Algorithm

• Lock the Priority Queue to pop its top sequence

![Diagram of Priority Queue]

• Lock the sequence to peek the next Task to execute

![Diagram of Sequence]
An overview of task scheduling in Chromium

Scheduling Algorithm

• Release the sequence lock and execute the task

• Lock the sequence and remove the task that was just executed
An overview of task scheduling in Chromium

Scheduling Algorithm

• If (SequenceLength != 0), lock the PriorityQueue and insert the sequence into it
Chromium processes

• The Browser Process (**CrBrowserMain**)
  • Receiving input from the OS, controls the browser UI (e.g. omnibox, back buttons, the tabstrip, menus, etc)

• Renderer Processes (**CrRendererMain**)
  • Putting each tab in its own render process

• The GPU Process (**CrGpuMain**)
  • GPU accelerated operations are issued to the graphics driver
How pages and images are loaded from the network into the renderer?
Chromium tracer

Features and specifications

- Tracing the system at user level to provide a call stack view
- Low overhead tracer with configurable parameters
- Ease of use
- Lack of information about kernel level
Multi-level analysis of chromium

Export Chromium events to Lttng

- Lttng tracepoints built into a shared library
- Developed API for exporting Chromium events to Lttng via the shared library
  - `dlopen("lib.so", RTLD_NOW)`
  - `dlsym(var, "[symbol]")`
- Getting kernel and user space events together
Trace Compass views

ChromeTaskTrackerPerTask.xml

- Tracking the tasks by a unique ID from the start up to the end
Trace Compass views

**ChromeTaskTrackerByThread.xml**

- Tracking the different states of tasks on different threads
Trace Compass views

Critical path view to show the execution bottleneck
Chromium performance analysis

**GpuChannelHost::Send** blocks UI thread to send IPC messages using **IPC::SyncChannel** (Issue 125264)
Chromium performance analysis

Processing the IPC message ViewHostMsg_ClosePage_ACK on the main thread can be too slow ([Issue 892747](#))
Conclusion

• Overview of task modeling
• Introducing the multi-process architecture of Chromium
• Proposing an approach to extract more information using LTTng, in order to analyze Chromium in more details
• Critical path analysis to detect performance bugs and latencies
Future work

• Analyzing the navigation to detect the root cause of some performance issues
• Building a tool to understand the interaction of tasks for a specific user action
• Making a more flexible and automated tool which is able to analyze Chromium
References

[1] Browser I/O Scheduler, URL: 
https://docs.google.com/document/d/1S2AAeoo1xa_vsLbDYBsDHCqhrkfiMgolPlyRi6kxa5k/edit#

[2] TaskTracker in SequenceManager,  
URL:https://docs.google.com/document/d/1sb5PdWz5q2pSZEU1mtR8lei9K-fjtpzkLmjRGGDfAPo/edit?ts=5ba16d8c#

Questions?

Majid.rezazadeh@polymtl.ca