ComputeCpp Profiler

Profiling ComputeCpp Applications

Objective of this document

ComputeCpp Professional Edition has built-in support to output a JSON file that contains profiling information of the runtime and the underlying OpenCL implementation. This JSON file is conformant with the Chrome Trace Events format and can be loaded in any Chrome installation by accessing the URL chrome://tracing.

The objective of this document is to describe how to activate profiling for a ComputeCpp application using the configuration file.

Enabling the JSON profiler backend

A configuration file may be used to configure how the ComputeCpp runtime will perform certain operations.

In order to enable profiling of the runtime, create a file and add the option `enable_profiling = true`. Point an environment variable called `COMPUTECPP_CONFIGURATION_FILE` to this file and run the application. The following is a single command to create a file called `sycl_config.txt` and export the environment variable to enable profiling:

- Windows:

  ```
  echo "enable_profiling = true" > sycl_config.txt; set COMPUTECPP_CONFIGURATION_FILE=sycl_config.txt
  ```

- Linux:

  ```
  echo "enable_profiling = true" > sycl_config.txt; export COMPUTECPP_CONFIGURATION_FILE=sycl_config.txt
  ```

Note that if `COMPUTECPP_CONFIGURATION_FILE` points to a file that doesn't exists, the runtime will fail to initialize.

Configuring the profiler

The following is a list of all the configuration options available to change how the profiler backend behaves:
<table>
<thead>
<tr>
<th>Configuration Option</th>
<th>Type</th>
<th>Default Value</th>
<th>Supported Backends</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>enable_profiling</td>
<td>boolean</td>
<td>false</td>
<td>all</td>
<td>Enables or disables profiling.</td>
</tr>
<tr>
<td>enable_kernel_profiling</td>
<td>boolean</td>
<td>true</td>
<td>json</td>
<td>Enables or disables profiling kernels running on a device. It will prevent injection of the property::queue::enable_profiling which could be useful when profiling some platforms.</td>
</tr>
<tr>
<td>profilingCollapse_transactions</td>
<td>boolean</td>
<td>false</td>
<td>json</td>
<td>Enabling this will cause all states of a transaction to be collapsed into a single entry. For long running applications this can be useful to reduce the size of the json file.</td>
</tr>
<tr>
<td>enable_buffer_profiling</td>
<td>boolean</td>
<td>true</td>
<td>json</td>
<td>Disabling this will prevent the json profiler to capture events on buffer. This can be useful when the application creates a large number of buffers and do not reuse them. It simplifies the json and allows it to be loaded more quickly.</td>
</tr>
</tbody>
</table>

**Profiling output**

By default, when the application finishes, the runtime will write the JSON file in the current working directory, usually the same directory as the binary of the application, in the format [executable_name]_[current_date].json.

This behaviour can be changed by setting the environment variable COMPUTECPP_PROFILING_OUTPUT. If this is set, the runtime will use the value of this variable as the output file. The file doesn't need to exist but the application must have permissions to create and write it.

**Performance Counters for Intel GPUs**

When running ComputeCpp applications in Intel GPU's that match one of the following architectures, the profiler can display performance counters for the ComputeBasic metrics set.

- Intel(R) Processors with Gen11 graphics devices (formerly Icelake),
- Intel(R) Processors with Gen9 graphics devices (formerly Skylake, Kaby Lake, Apollo Lake/Broxton, Gemini Lake, Coffee Lake),
- Intel(R) Processors with Gen8 graphics devices (formerly Broadwell),
- Intel(R) Processors with Gen7.5 graphics devices (formerly Haswell).

No extra configuration is required, as long as the system supports Intel's Metrics Discovery API.

**Example**

The following is an example of the Google Chrome Visualizer that displays a execution of **BabelStream** in an Intel GPU.

Note at the bottom the performances counters. Many more counters are available and they can be accessed by scrolling the window.