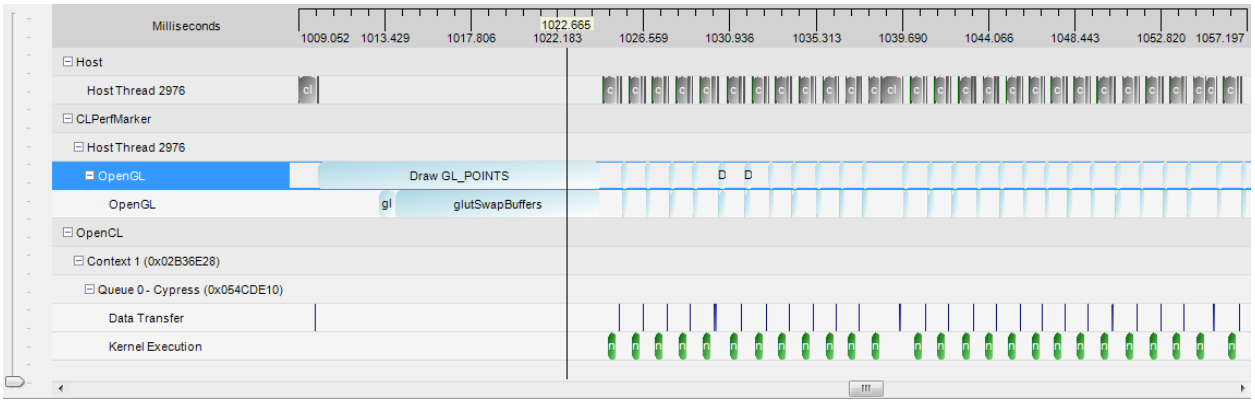
AMD APP Profiler OpenCL Performance Marker API

v1.1

Last updated: 14 Nov, 2011

Introduction



The OpenCL Performance Marker (CLPerfMarker) library provides a simple host-code instrumentation API that can help users to analyze their OpenCL applications. It allows users to instrument their code with calls to clBeginPerfMarkerAMD() and clEndPerfMarkerAMD(). These calls are then used by the APP Profiler to annotate the host-code timeline in a hierarchical way.

Each CLPerfMarker has a name, two timestamps which define the beginning and end of the host-code that the CLPerfMarker encloses and an optional group name. In the AMD APP Profiler Timeline view, markers from different threads are grouped separately under dedicated thread branches. All top-level markers are under a thread branch. There are two ways to create sub-branches under a thread branch. The first way is to make nested calls to clBeginPerfMarkerAMD (). The second way is to create a top-level marker with a group name. Group names are ignored if markers are not directly under a thread branch.

All APIs are thread-safe.

Function Definitions

clInitializePerfMarkerAMD()

Syntax:

int clInitializePerfMarkerAMD()

Description:

This function initializes the CLPerfMarker library. It checks whether or not the APP Profiler is enabled, and if not, it returns AP_APP_PROFILER_NOT_DETECTED. Depending on the APP Profiler API trace mode being used, CLPerfMarker may cache all the trace results and flush them to disk when clFinalizePerfMarkerAMD() is called or it may flush trace result periodically and merge them when clFinalizePerfMarkerAMD () is called.

Returns:

AP_SUCCESS	Returned if CLPerfMarker was successfully initialized.
AP_FINALIZED_PERF_MARKER	Returned if clFinalizePerfMarker has already been called.
AP_APP_PROFILER_NOT_DETECTED	Returned if the application has not been run with the AMD APP Profiler.

clBeginPerfMarkerAMD()

Syntax:

```
int clBeginPerfMarkerAMD(
        const char* szMarkerName,
        const char* szGroupName);
```

Description:

This function marks the beginning of a CLPerfMarker. If CLPerfMarker has not been initialized, AP_UNINITIALIZED_PERF_MARKER is returned. If the call is successful, a timestamp is internally recorded. The specified marker name is displayed on the timeline block. A nested clBeginPerfMarkerAMD() call creates a new sub-branch with its group name ignored. A top-level clBeginPerfMarkerAMD() call with a group name specified also creates a new branch.

Returns:

AP_SUCCESS	Returned if operation succeeded.
AP_UNINITIALIZED_PERF_MARKER	Returned if clInitializePerfMarkerAMD() has not been called.
AP_FINALIZED_PERF_MARKER	Returned if clFinalizePerfMarker() has been called.

AP_NULL_MARKER_NAME	Returned if szMarkerName is equal to NULL.

clEndPerfMarkerAMD()

Syntax:

int clEndPerfMarkerAMD();

Description:

This function marks the end of a CLPerfMarker. If CLPerfMarker has not been initialized, AP_UNINITIALIZED_PERF_MARKER is returned. If the call is successful, a timestamp is recorded. If clEndPerfMarkerAMD() is called without sufficient clBeginPerfMarkerAMD() calls, AP_UNBALANCED_MARKER is returned.

Returns:

AP_SUCCESS	Returned if operation succeeded.
AP_UNINITIALIZED_PERF_MARKER	Returned if clInitializePerfMarkerAMD() has not been called.
AP_FINALIZED_PERF_MARKER	Returned if clFinalizePerfMarker() has been called.
AP_UNBALANCED_MARKER	Returned if insufficient clBeginPerfMarkerAMD() calls have been made to balance this call.

clFinalizePerfMarkerAMD()

Syntax:

int clFinalizePerfMarkerAMD()

Description:

This function finalizes CLPerfMarker. Unless this function is called, no CLPerfMarker file will be generated. After this call, no further CLPerfMarker will be recorded. The CLPerfMarker output file name is defined as the base output file name specified when running APP Profiler plus .clperfmarker extension.

Returns:

AP_SUCCESS	Returned if CLPerfMarker was successfully finalized.
AP_UNINITIALIZED_PERF_MARKER	Returned if CLPerfMarker has not been initialized.

AP_FAILED_TO_OPEN_OUTPUT_FILE	Returned if CLPerfMarker failed to write to output file.