

trace调研 --11.23

trace_profiler.py

核心的类是 `Trace` :

初始化时完成tasks, tasks_list, locations, threads_stack, pending_files, filename对应文件的导入。

其中类 `TraceTask` 负责记录threadID, taskID, locationID, parentThreadID, parentTaskID, beginTimestamp, endTimestamp, totalTimeIPP, selfTimeIPP, totalTimeOpenCL, selfTimeOpenCL。每个thread和task都有对应的ID, 这里还记录了其父thread和父task的ID。Timestamp是记录时长的时间戳, TimeIPP和TimeOpenCL 是使用IPP 和OpenCL的次数。

类 `TraceLocation` 负责记录 locationID, filename, line, name, 这个在行处理时会用到。

`parse_file(filename)` 函数负责文件处理, 主要任务是将文件按行分割, 识别#thread file, 将其放入opening_files的队列中。接着进行行处理。

`parse_line(line)` 负责行处理, 主要任务是将行按, 分割, 分割后的每一块为opts, 分为l,b,e开头三种情况, b开头时特殊处理task和thread的相关ID和队列, e开头时特殊处理TimeIPP和TimeOpenCL。

`load(filename)` 主要任务是将filename对应的文件进入队列, pop pendingfile 并parse。

`process()` 主要将task按开始时间排序, 计算totaltime和parallel的total time。

`dump(max_entries)` 主要任务是从tasks_list 中call max_entries个task并处理。

trace.cpp

一个重要的类是 `TraceManager` . 以其定义的TraceManagerThreadLocal 就是我们的 `ctx` .

任何基本操作时首先会检查是否用 IPP / OPENCL/ OPENVX 。

最重要的大类是**Region**:

其中重要的数据结构和参数的含义:

```

struct LocationExtraData;    //location的StaticStorage的信息
struct LocationStaticStorage
{
    LocationExtraData** ppExtra;    //< implementation specific data
    const char* name;              //< region name (function name or
other custom name)
    const char* filename;          //< source code filename
    int line;                      //< source code line
    int flags;                    //< flags (implementation code path:
Plain, IPP, OpenCL)
};
class Impl; //if region is active
int implFlags; // see RegionFlag, 0 if region is ignored

```

重要的函数：

`enterRegion(TraceManagerThreadLocal& ctx)` 主要任务是将该region设置为active状态，修改region深度，建立对应的TraceStorage. `leaveRegion(TraceManagerThreadLocal& ctx)` 主要任务是在任务结束时记录任务时长，修改调用深度。在用了IPP / OPENCL/ OPENVX 还有对应的特殊处理。

`registerRegion(TraceManagerThreadLocal& ctx)` 主要任务是在用了ITT的情况下，将该任务的threadID记录并建立ITT ID.

`Region::destroy()` 与`leaveRegion`有相似之处，但多了 `pImpl` 的处理。

备记：

IPP: Integrated Performance Primitives.

ITT: the ITT API is used to annotate a user's program with additional information that can be used by correctness and performance tools.