Evaluation task for Awkward Array GSoC project

Pratyush Das[1]

[1]Institute of Engineering & Management, Kolkata

Abstract

In this report, we demonstrate a sample GPU kernel designed to be an alternative to the CPU backend for array operations in the Awkward Array project. In particular, we implement a CUDA translation of the *awkward_listarray_compact_offsets* CPU kernel, and show how parallelizing the code on a GPU could significantly increase the speed of computation.

1 Introduction

The sample CPU kernel (directly taken from the Awkward Array codebase) to be translated is defined below -

```
template <typename C, typename T>
ERROR awkward_listarray_compact_offsets(T* tooffsets,
    const C* fromstarts, const C* fromstops, int64_t
    startsoffset, int64_t stopsoffset, int64_t length) {
  tooffsets[0] = 0;
  for (int64_t i = 0; i < length; i++) {
    C start = fromstarts[startsoffset + i];
    C stop = fromstops[stopsoffset + i];
    if (stop < start) {
      return failure(”stops[i] < starts[i]”, i,
          kSliceNone);
    }
    tooffsets[i + 1] = tooffsets[i] + (stop - start);
  }
  return success();
}
```

Translating this particular CPU kernel serves as good test because parallelizing this CPU kernel involves overcoming the loop carried dependency in the above code -

```
tooffsets[i + 1] = tooffsets[i] + (stop - start);
```

where *tooffsets[i + 1]* depends on *toffsets[i]*.