Profiling of Parallel Applications

Jakub Beránek (author), Petr Gajdoš (supervisor)

Motivation

- Provide insight into runtime behaviour of CUDA threads
- Help programmers optimize their code by discovering hidden bottlenecks caused by non-optimal memory accesses
- Design an easy to use CUDA instrumentation framework - current tools are difficult to set up and break with new framework releases
- Record all accesses - current tools provide only aggregated data

Contribution

- CUDA profiling library that **records memory accesses**
- Analytical and **visualization** tool
- Key features:
  - detailed thread memory access recording
  - shared memory conflict detection
  - visualization of memory access patterns
  - simple setup (just include a single header file)
  - independent of CUDA SDK version

Implementation

- LLVM plugin that finds memory accesses in CUDA functions
- Accesses are wrapped with code that records their value, size, address, data type and thread ID at runtime
- GPU allocations are tracked to provide detailed address space information
- Accesses are exported to JSON or Protobuf files and can be optionally compressed
- Recorded data is visualized in a React web app, which provides filtering of accesses and displays shared memory conflicts and memory access strides

SOURCE CODE

```
1 int val = data[threadIdx.x];
2 if (threadIdx.y == 0) {
3     shared[threadIdx.x] = val;
4 }
```

INSTRUMENTATION

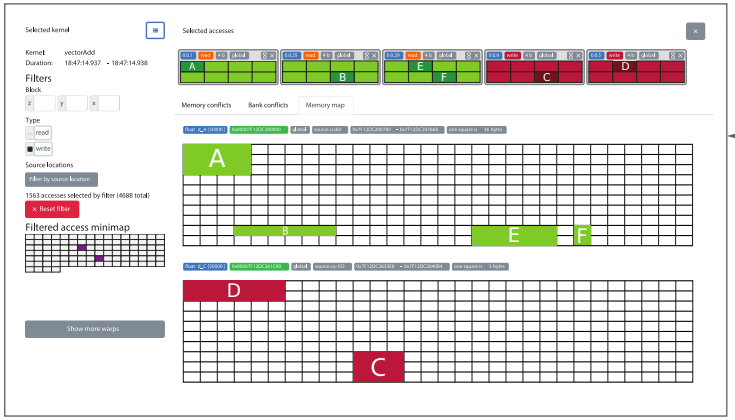
```
1 __cu_load(
2     data + threadIdx.x,
3     sizeof(int),
4     "int",
5     "kernel.cu:42",
6     data[threadIdx.x]
7 );
```

CLANG COMPILER

JSON

```
1  [{
2    thread: {x: 0, y: 1},
3    type: "write",
4    value: 0x42,
5    address: 0xFFFF1234
6  }, {
7    thread: {x: 0, y: 2},
8    type: "write",
9    value: 0x38,
10   address: 0xFFFF1238
11 }, {
12   thread: {x: 0, y: 3},
13   type: "read",
14   value: 0x27,
15   address: 0xFFFF123B
16 }]
```

USER INTERFACE



Conclusion

- Memory access anomalies become obvious when visualized
- Runtime behaviour inspection is useful for memory optimization experiments
- Working open-source tool tested on official CUDA samples
- Created library also serves as a maintainable CUDA instrumentation framework
- Could be combined with CPU access recording and integrated into an IDE

https://github.com/kobzol/cuda-profile