Sanajeh: A DSL for GPGPU programming with Python objects

東京工業大学 Chen Xin JiZhe, 増原 英彦, Matthias Springer

Proposal

- Provide SMMO programming model in python
- Make it easier to write the program compared to using DynaSOAr APIs

Previous work

DynaSOAr\cite{1}

- A dynamic SOA(Structure Of Array) data layout memory allocator written in C++/CUDA
- Based on the SMMO (Single-Method Multiple-Objects) model, which realizes parallelism by running one single method on all objects of a class.

```python
class A{
    int index = 0;
    __device__ void addIndex() {
        index += 1;
    }
}
int main(){
    //Let A has 3 objects with index 0, 1, 2
    allocator_handle->parallel_do<A, &addIndex>();
    //A will have 3 objects with index 1, 2, 3
}
```

Fig 1. Example of SMMMO

Problems in DynaSOAr

- Requires programmers to write codes in an extended syntax.
  For example the “__device__” keyword marked as red in Fig 1.
- Difficult for programmers to write complex benchmark.
  For example the Barnes-Hut simulation.

Methods

1. A core language based on SMMMO
   - Written in normal python syntax.
   - Create objects of a class on device by new_all (or new__ when parameters is needed) function
   - Detect and mark device data automatically, they will be compiled into CUDA code which will be run by DynaSOAr

2. Provide a library for high-level parallel programming patterns
   - Including parallel tree and graph algorithms
   - Hide difficult implementation for example the race among different threads

"Fig 3 is a step in Barnes-Hut simulation, in which the Race Condition(concurrent data modifying among threads) will happen"

Reference