Ideas for Low Celluclast Performance

13.11.2019

Basic Architecture

- Accesses different memory than CPU
- Starting thousands of threads at low costs

Memory Management

```
typedef int dtype;
class Storage {
   public:
    explicit Storage(const std::vector<int>&);

   private:
    std::vector<int> _data;
    dtype* _cpu_pointer;
    dtype* _gpu_pointer;
    void initialize_gpu_memory();
};
```

- Memory pool, takes ownership
- Initializes the gpu memory as copy
- two differetn pointer, to cpu/gpu locations

```
typedef int dtype;
class Storage {
    public:
     explicit Storage(const std::vector<int>&);
     const dtype* cpu_pointer_const();
     const dtype* gpu_pointer_const();
     dtype* cpu_pointer();
     dtype* gpu_pointer();

    private:
     std::vector<int> _data;
     dtype* _cpu_pointer;
     dtype* _gpu_pointer;
     void initialize_gpu_memory();
     std::string recent_head;
     void sync_to_cpu();
     void sync_to_gpu();
};
```

- accesses pointers

Merge

Write about CPU merge

How to spwan to many threads?

Paper that does that, show as example, the cutting approach
Naiv approach : 2 Threads Cut both a and b into half,

$$A = 0000 B \qquad\qquad = 1111 C = ????????$$

$$A = \underbrace{00}_{\text{Thread 1}} \mid \underbrace{00}_{\text{Thread 2}} B = \underbrace{11}_{\text{Thread 1}} \mid \underbrace{11}_{\text{Thread 2}} C = \underbrace{????}_{\text{Thread 1}} \mid \underbrace{????}_{\text{Thread 2}}$$

Merge