Intermediate Report for ME759

Lu Duan

Milestone:

- Present profiling result for serial computing algorithm
- Propose parallel part suggestion and possible implementations

Progress update

1. Profiling

Basically, there are three functions called to compute the hash value of any given strings:

- MD5Init()
- MD5Update()
- MD5Final()

The major part of computing is done by a subfunction MD5Transform() inside the MD5Update(). There are also a bunch of memory copy/set before and after each MD5Transform() inside MD5Update() and MD5Final(). So, from the structure, it's easy to guess and verify that MD5Transform() takes the majority of the program's running time. Here is one example of the profiling result:

```
MD5Init runtime = 0.23 us
Total transform time = 0.177 us
MD5 memcpy in update = 0.392 us
MD5Update runtime = 19.851 us
Total transform time = 0.161 us
MD5 memcpy in update = 1.134 us
MD5 memset in transform = 2.292 us
Single transform time = 14.215 us
Total transform time = 29.61 us
MD5 memcpy in update = 0.25 us
MD5 memset in final = 2.113 us
MD5Final runtime = 120.077 us

MD5 of '1234567' = 6bd12a1de6390cb51a600806ffffcfc1
Total runtime = 270.611 us
```

2. Improvement Proposal

First, we could parallelize and offload the three rounds of bit manipulation inside MD5Transform() to GPU. Memory copy involves only a short length of string which makes the calculation computing-bound.

Second, there are encode(), decode(), memcpy(), memset() in the MD5Final(), which all include a giant for loop. We could parallelize this part as well.