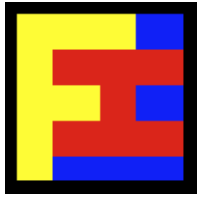


FOUNTAINHEAD

Programming GPUs with CUDA Thrust

~ A Rapid Development Framework for GPU ~

Baruch College MFE “Big Data in Finance” Course
30th January 2014



FOUNTAINHEAD

What is “Thrust?”

Library modeled on the Standard Template Library (STL) of C++

Provides a layer of abstraction on top of CUDA

Improves programmer productivity (both skilled & unskilled)

Retains high-performance

Production ready



FOUNTAINHEAD

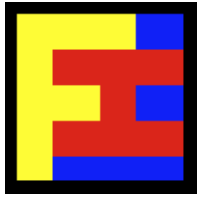
What does it contain?

Vectors

Iterators

Algorithms

(Tip: The header files reveal all.)



FOUNTAINHEAD

Start using Thrust

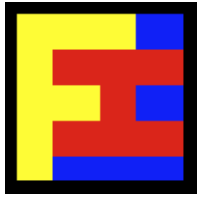
Uses existing CUDA tool chain

If you are set up to develop with CUDA, you can develop with Thrust

Download Thrust code

`#include` in .cu files

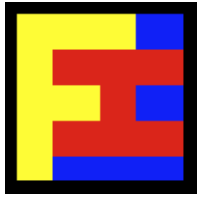
No separate build, or library to link with



FOUNTAINHEAD

Thrust code example

```
int main(void)
{
    thrust::host_vector<int> h_vec(1 << 24);
    thrust::generate(h_vec.begin(), h_vec.end(), rand);
    thrust::device_vector<int> d_vec=h_vec;
    thrust::sort(d_vec.begin(), d_vec.end());
    thrust::copy(d_vec.begin(),
                 d_vec.end(),
                 h_vec.begin());
    return 0;
}
```



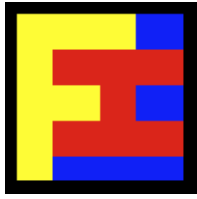
FOUNTAINHEAD

Code example key points

More compact than CUDA runtime or driver API code

Code on host and device is similar

Parallelism and housekeeping hidden from the programmer



FOUNTAINHEAD

Thrust functionality

Iterators and static dispatching

Transformations

Reductions

Prefix-sums (scan operations)

Reordering

Sorting



FOUNTAINHEAD

Thrust functionality (cont.)

Fancy iterators

`constant_iterator`

`counting_iterator`

`transform_iterator`

`permutation_iterator`

`zip_iterator`



FOUNTAINHEAD

Code organization: includes

Functionality split across a large number of header files

Well named

Reading the source is informative (getting into the mind of the author)



FOUNTAINHEAD

Thrust namespace members

Typedefs

Functions

(Remember, only two containers:

`host_vector`

and

`device_vector`)