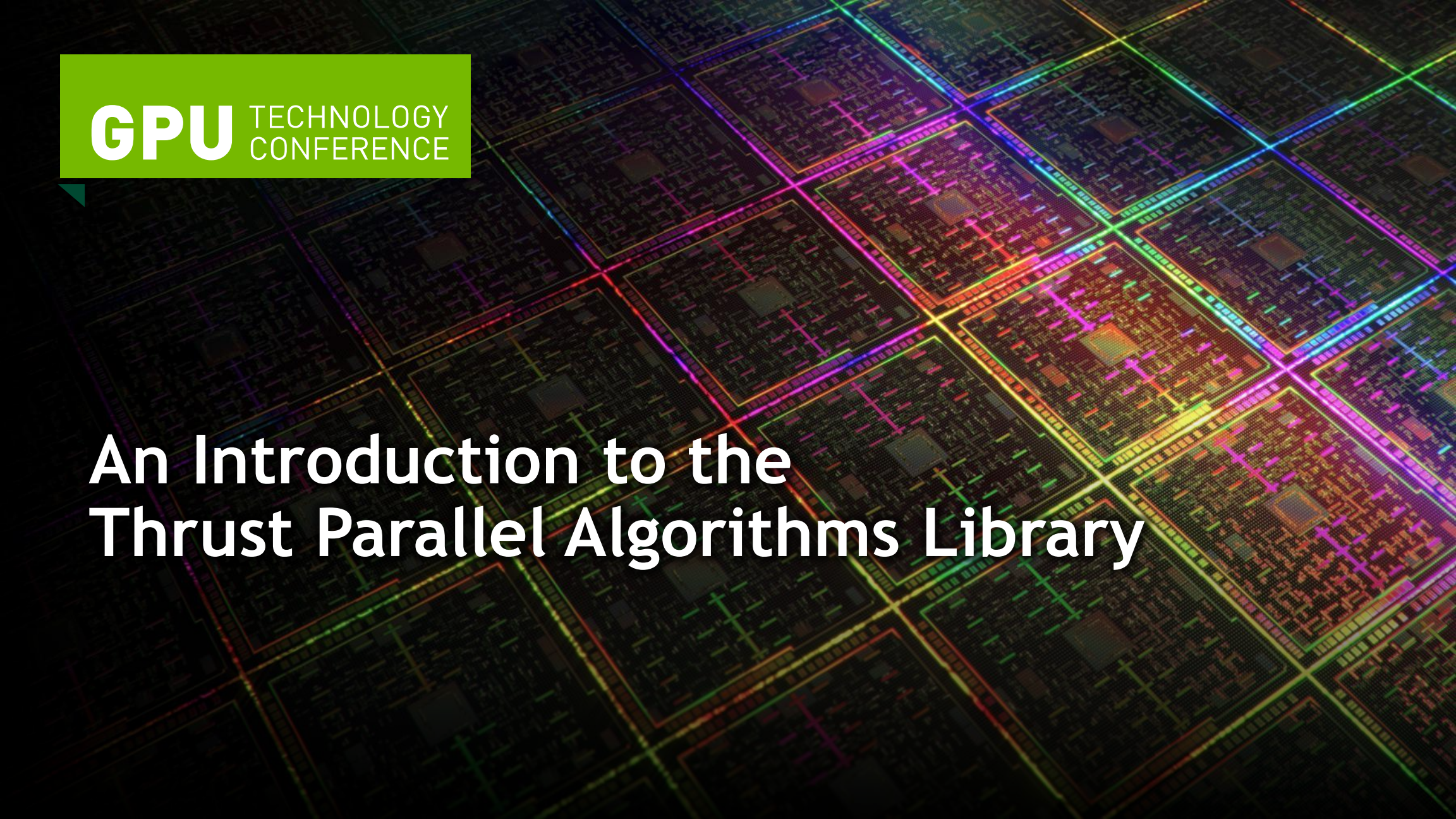
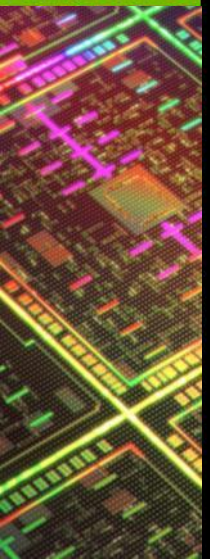
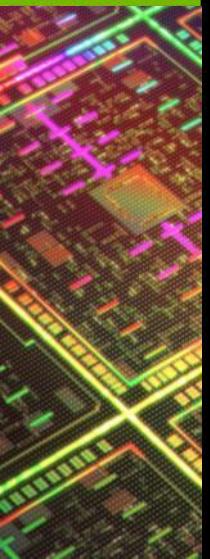


An Introduction to the Thrust Parallel Algorithms Library



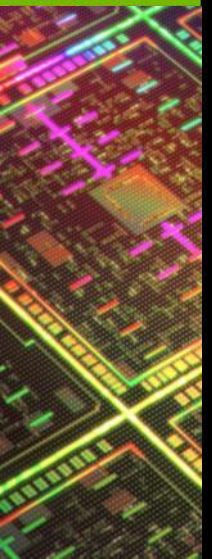
What is Thrust?

- High-Level Parallel Algorithms Library
- Parallel Analog of the C++ Standard Template Library (STL)
- Performance-Portable Abstraction Layer
- Productive way to program CUDA



Example

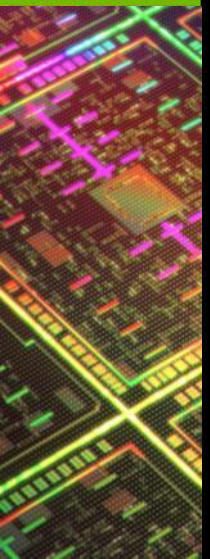
```
#include <thrust/host_vector.h>
#include <thrust/device_vector.h>
#include <thrust/sort.h>
#include <cstdlib>

int main(void)
{
    // generate 32M random numbers on the host
    thrust::host_vector<int> h_vec(32 << 20);
    thrust::generate(h_vec.begin(), h_vec.end(), rand);

    // transfer data to the device
    thrust::device_vector<int> d_vec = h_vec;

    // sort data on the device
    thrust::sort(d_vec.begin(), d_vec.end());

    // transfer data back to host
    thrust::copy(d_vec.begin(), d_vec.end(), h_vec.begin());

    return 0;
}
```



Easy to Use

- Distributed with CUDA Toolkit
- Header-only library
- Architecture agnostic
- Just compile and run!

```
$ nvcc -O2 -arch=sm_20 program.cu -o program
```



Why should I use Thrust?

Productivity

- Containers
  - `host_vector`
  - `device_vector`
- Memory Mangement
  - Allocation
  - Transfers
- Algorithm Selection
  - Location is implicit

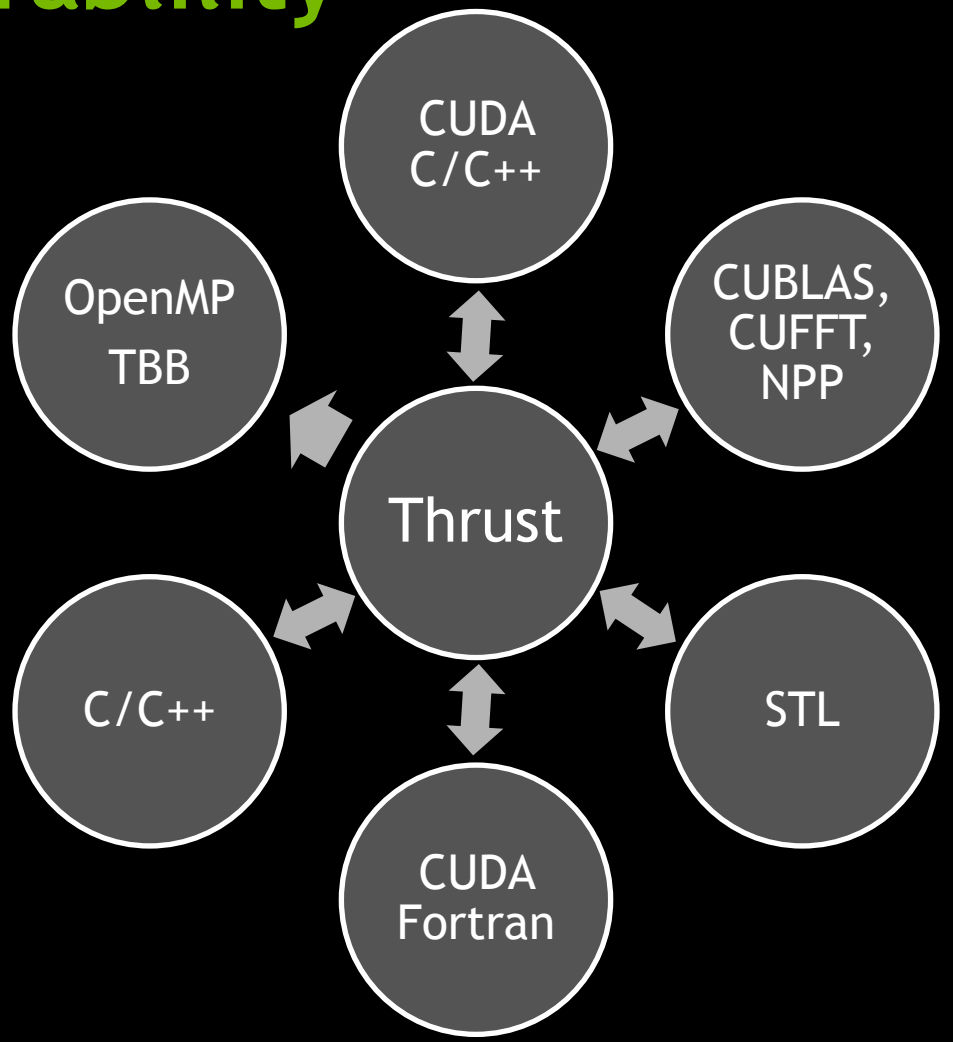
```
// allocate host vector with two elements
thrust::host_vector<int> h_vec(2);

// copy host data to device memory
thrust::device_vector<int> d_vec = h_vec;

// write device values from the host
d_vec[0] = 27;
d_vec[1] = 13;

// read device values from the host
int sum = d_vec[0] + d_vec[1];

// invoke algorithm on device
thrust::sort(d_vec.begin(), d_vec.end());

// memory automatically released
```

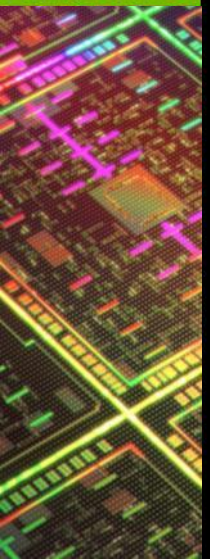
Productivity

- Large set of algorithms
  - ~75 functions
  - ~125 variations
- Flexible
  - User-defined types
  - User-defined operators

Algorithm	Description
`reduce`	Sum of a sequence
`find`	First position of a value in a sequence
`mismatch`	First position where two sequences differ
`inner_product`	Dot product of two sequences
`equal`	Whether two sequences are equal
`min_element`	Position of the smallest value
`count`	Number of instances of a value
`is_sorted`	Whether sequence is in sorted order
`transform_reduce`	Sum of transformed sequence

Interoperablility





Portability

- Support for CUDA, TBB and OpenMP
  - Just recompile!

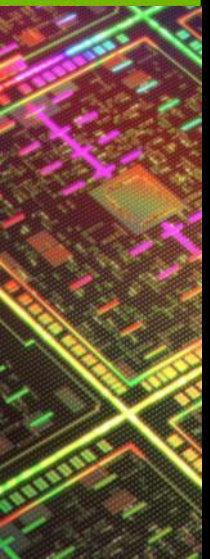
```
nvcc -DTHRUST_DEVICE_SYSTEM=THRUST_HOST_SYSTEM_OMP
```

NVIDA GeForce GTX 580

```
$ time ./monte_carlo
pi is approximately 3.14159

real    0m6.190s
user    0m6.052s
sys 0m0.116s
```

Intel Core i7 2600K

```
$ time ./monte_carlo
pi is approximately 3.14159

real    1m26.217s
user    11m28.383s
sys 0m0.020s
```



Backend System Options

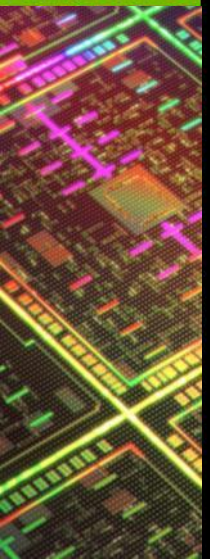
Host Systems
THRUST_HOST_SYSTEM_CPP
THRUST_HOST_SYSTEM_OMP
THRUST_HOST_SYSTEM_TBB

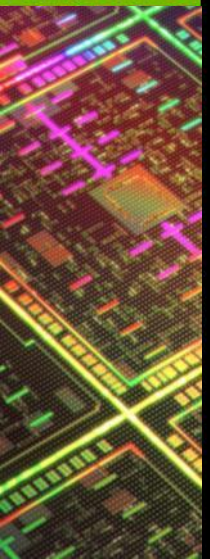
Device Systems
THRUST_DEVICE_SYSTEM_CUDA
THRUST_DEVICE_SYSTEM_OMP
THRUST_DEVICE_SYSTEM_TBB

Multiple Backend Systems

- Mix different backends freely within the same app

```
thrust::omp::vector<float> my_omp_vec(100);
thrust::cuda::vector<float> my_cuda_vec(100);

...

// reduce in parallel on the CPU
thrust::reduce(my_omp_vec.begin(), my_omp_vec.end());

// sort in parallel on the GPU
thrust::sort(my_cuda_vec.begin(), my_cuda_vec.end());
```





Thrust_sort

- Open exercises/cuda/thrust_sort/kernel.cu
- Code should build without modification
- What is performance of GPU versus CPU code?

Potential Workflow

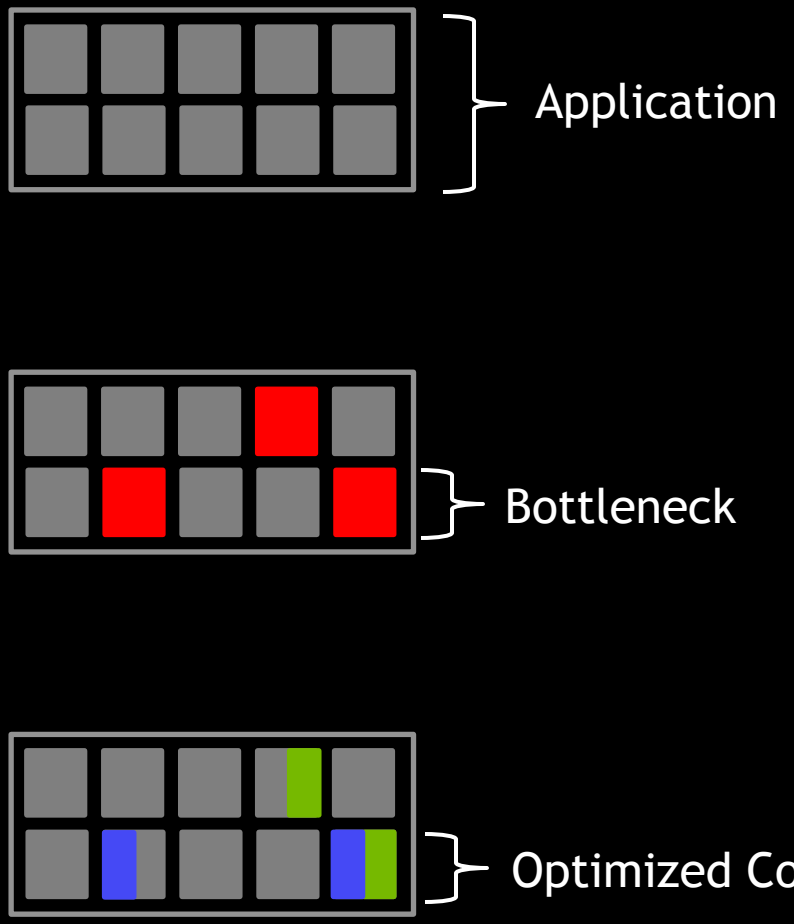
Thrust Implementation

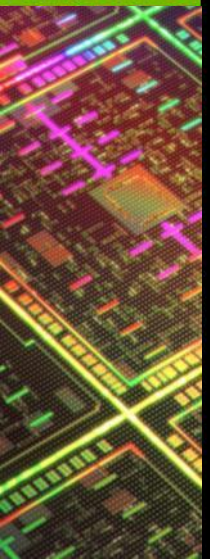
↓

Profile Application

↓

Specialize Components





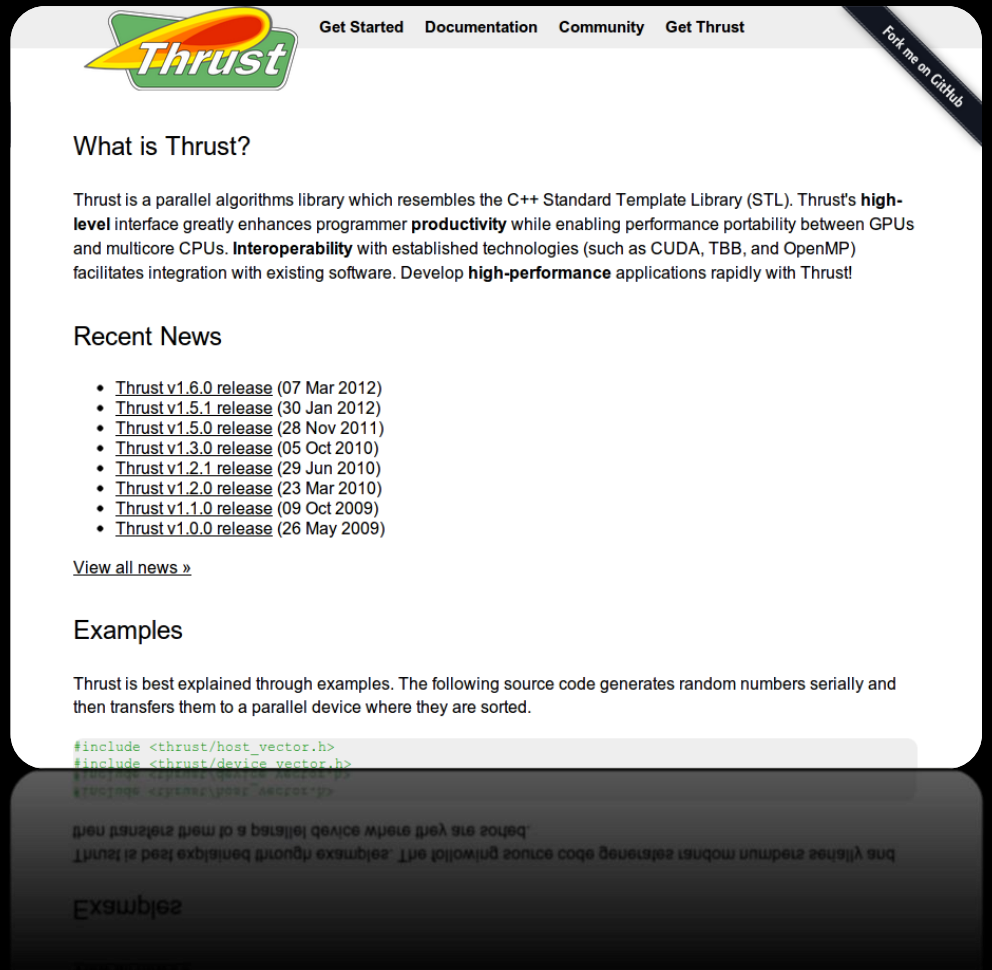
Robustness

- Reliable
  - Supports all CUDA-capable GPUs
- Well-tested
  - ~850 unit tests run daily
- Robust
  - Handles many pathological use cases

Openness

- Open Source Software
  - Apache License
  - Hosted on GitHub
- Welcome to
  - Suggestions
  - Criticism
  - Bug Reports
  - Contributions

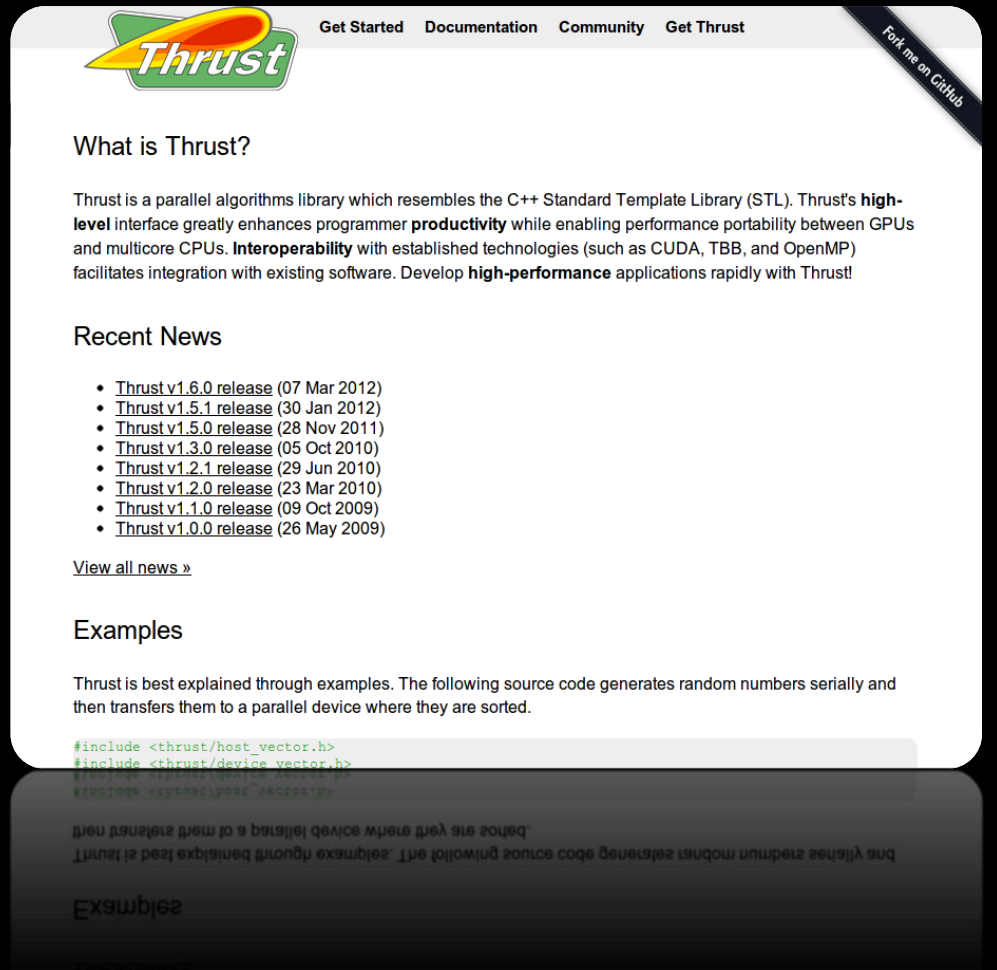
thrust.github.com



Resources

- Documentation
- Examples
- Mailing List
- Webinars
- Publications

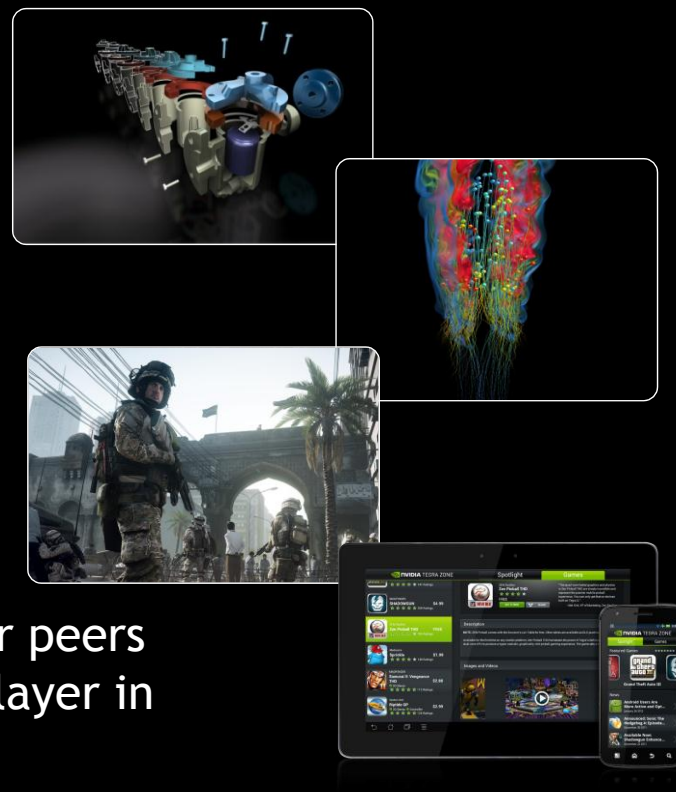
thrust.github.com



GPU Technology Conference 2013
March 18-21 | San Jose, CA

Why attend GTC?

GTC advances global awareness of the dramatic changes we're seeing in science and research, graphics, cloud computing, game development, and mobile computing, and how the GPU is central to innovation in all areas.



Ways to participate

- Submit a Research Poster – share your work and gain exposure as a thought leader
- Register – learn from the experts and network with your peers
- Exhibit/Sponsor – promote your organization as a key player in the GPU ecosystem

Visit www.gputechconf.com