Princeton GPU Hackathon

EcoSLIM

Jun Zhang, U of Arizona
Hoang Tran, Princeton U
Chen Yang, Princeton U

Mentors
Troy Comi, Princeton U
Carl Ponder, NVIDIA

Goals

	P(:,1)	P(:,2)	P(:,3)	P(:,4)	P(:,5)	P(:,6)	P(:,7)	P(:,8)	....
1								1	
2								0	
3								0	
4								1	
5								1	
6								0	
7								1	
8								1	
9								0	
10								1	
....									

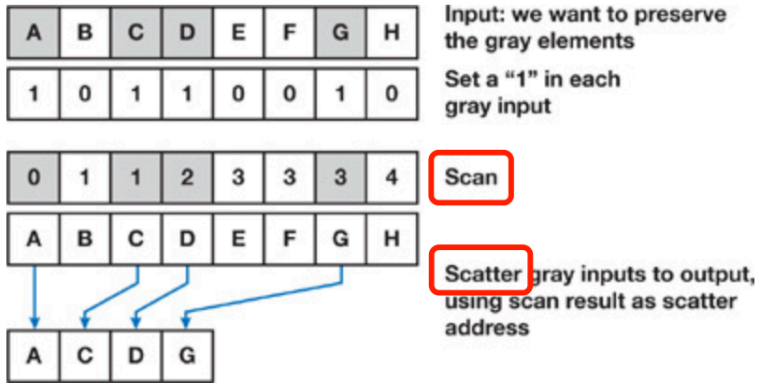

Figure 39-10 Scan and Scatter

```
template <class T, bool isBackward>
__global__ void compactData(T                  *d_out,
                            size_t             *d_numValidElements,
                            const unsigned int *d_indices, // Exclusive Sum-Scan Result
                            const unsigned int *d_isValid,
                            const T            *d_in,
                            unsigned int       numElements)
{
    if (threadIdx.x == 0)

if (iGlobal < numElements && d_isValid[iGlobal] > 0) {
    d_out[d_indices[iGlobal]] = d_in[iGlobal];
}
```

Scatter in cudpp

1D array

1) For scan: writing a wrapper to call **exclusive_scan in Thrust** which is a CUDA C++ template library;
2) For scatter: modifying the kernel from cudpp to handle 2D particle array P;
3) For scatter: writing a wrapper to call this CUDA C++ kernel in Fortran; Or probably rewrite it in CUDA Fortran;
4) Test and profile scan and scatter using restart files generated in previous simulations;
5) Using cmake, Nsight, etc.

Wrapper Functions

thrust_module.cuf | cscan.cu 9+

chenyang > Desktop > GPU_Hackathon_Princeton > test > thrust_scan > cscan.cu

```
#include <thrust/host_vector.h>
#include <thrust/device_vector.h>
#include <thrust/scan.h>
extern "C" {
    void scan_int_wrapper( int *data_in, int N, int *data_out)
    {
    thrust::device_ptr<int> dev_ptr_in(data_in);
    thrust::device_ptr<int> dev_ptr_out(data_out);
    thrust::exclusive_scan(dev_ptr_in, dev_ptr_in+N, dev_ptr_out);
    }
    void scan_float_wrapper( float *data_in, int N, float *data_out)
    {
    thrust::device_ptr<float> dev_ptr_in(data_in);
    thrust::device_ptr<float> dev_ptr_out(data_out);
    thrust::exclusive_scan(dev_ptr_in, dev_ptr_in+N, dev_ptr_out);
    }
    void scan_double_wrapper( double *data_in, int N, double *data_out)
    {
    thrust::device_ptr<double> dev_ptr_in(data_in);
    thrust::device_ptr<double> dev_ptr_out(data_out);
    thrust::exclusive_scan(dev_ptr_in, dev_ptr_in+N, dev_ptr_out);
    }
}
```

efile | thrust_module.cuf | cscan.cu 9+

chenyang > Desktop > GPU_Hackathon_Princeton > test > thrust_scan > thrust_module.cuf > ...

```
module thrust
interface thrustscan

subroutine scan_int(input,N,output) bind(C,name="scan_int_wrapper")
use iso_c_binding
integer(c_int),device:: input(*)
integer(c_int),device:: output(*)
integer(c_int),value:: N
end subroutine

subroutine scan_float(input,N,output) bind(C,name="scan_float_wrapper")
use iso_c_binding
real(c_float),device:: input(*)
real(c_float),device:: output(*)
integer(c_int),value:: N
end subroutine

subroutine scan_double(input,N,output) bind(C,name="scan_double_wrapper")
use iso_c_binding
real(c_double),device:: input(*)
real(c_double),device:: output(*)
integer(c_int),value:: N
end subroutine

end interface
end module thrust
```

```
program testsort
   use thrust

   real, allocatable :: cpuData(:,:)
   real, allocatable, device :: gpuData(:,:)
   integer:: N,count

   N=6
   allocate(cpuData(N,2))
   allocate(gpuData(N,2))

   count=0

   do i=1,N
      do j=1,2
         count=count+1
         cpuData(i,j)=count
      end do
   end do

   write(*,'(a,6(f5.1,1x))') "Before sanning", cpuData(:,1)
   write(*,'(a,6(f5.1,1x))') "Before sanning", cpuData(:,2)

   gpuData=cpuData
   call thrustscan(gpuData(:,2),size(gpuData(:,2)),gpuData(:,2))
   cpuData=gpuData
   write(*,'(a,6(f5.1,1x))') "After sanning", cpuData(:,1)
   write(*,'(a,6(f5.1,1x))') "After sanning", cpuData(:,2)

end program
```

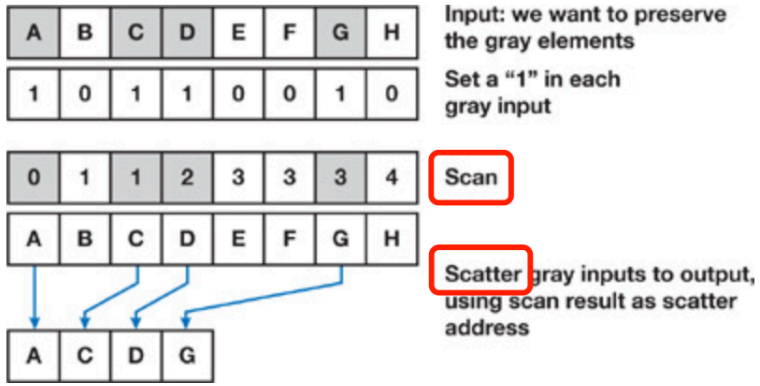

Figure 39-10 Scan and Scatter

Results

```
Currently Loaded Modulefiles:
 1) nvhpc/21.1   2) cudatoolkit/11.1   3) openmpi/
Before sanning  1.0   3.0   5.0   7.0   9.0  11.0
Before sanning  2.0   4.0   6.0   8.0  10.0  12.0
After sanning  1.0   3.0   5.0   7.0   9.0  11.0
After sanning  0.0   2.0   6.0  12.0  20.0  30.0
```

Column 2

Column 2

Column 2

Problems

- What problems are you currently facing?
  MPI failed

```
program testsort
    use thrust
    use mpi
    use mpiDeviceUtil

    real, allocatable :: cpuData(:,:)
    real, allocatable, device :: gpuData(:,:)
    integer:: N,count,deviceID
    integer:: rank, nproc, ierr

    call mpi_init(ierr)
    call mpi_comm_size(mpi_comm_world, nproc, ierr)
    call mpi_comm_rank(mpi_comm_world, rank, ierr)

    call assignDevice(deviceID)

    N=6; count=0
    allocate(cpuData(N,2))
    allocate(gpuData(N,2))

    do i=1,N
        do j=1,2
```

Assign GPU

```
Currently Loaded Modulefiles:
 1) nvhpc/21.1   2) cudatoolkit/11.1   3) openmpi/nvhpc-21.1/4.1.0
srun: error: della-i14g2: tasks 2-3: Illegal instruction (core dumped)
srun: launch/slurm: _step_signal: Terminating StepId=34826029.0
srun: error: della-i14g1: tasks 0-1: Illegal instruction (core dumped)
~
```