**Lab Exercise 9.3 – Develop and Execute Sample Programs Leveraging Other CUDA Libraries**

**Objective:**

* Explore additional CUDA libraries beyond cuBLAS and cuFFT.
* Learn to develop and execute programs using libraries like cuRAND (random number generation) and Thrust (high-level STL-like API for CUDA).
* Understand how these libraries accelerate development and performance.

**1. Using cuRAND – CUDA Random Number Generation Library**

**Goal:** Generate random numbers on the GPU using cuRAND.

#include <iostream>

#include <curand.h>

#define N 10

int main() {

float \*devData, \*hostData;

curandGenerator\_t gen;

// Allocate memory

cudaMalloc((void \*\*)&devData, N \* sizeof(float));

hostData = (float \*)malloc(N \* sizeof(float));

// Create generator

curandCreateGenerator(&gen, CURAND\_RNG\_PSEUDO\_DEFAULT);

curandSetPseudoRandomGeneratorSeed(gen, 1234ULL);

// Generate uniform floats on device

curandGenerateUniform(gen, devData, N);

// Copy to host

cudaMemcpy(hostData, devData, N \* sizeof(float), cudaMemcpyDeviceToHost);

std::cout << "Random numbers generated on GPU:\n";

for (int i = 0; i < N; ++i)

std::cout << hostData[i] << std::endl;

// Clean up

curandDestroyGenerator(gen);

cudaFree(devData);

free(hostData);

return 0;

}

**Compile and Run:**

nvcc -o curand\_example curand\_example.cu -lcurand

./curand\_example

**2. Using Thrust – High-Level CUDA Library**

**Goal:** Sort an array on the GPU using Thrust.

#include <thrust/host\_vector.h>

#include <thrust/device\_vector.h>

#include <thrust/sort.h>

#include <iostream>

int main() {

// Create host vector with random values

thrust::host\_vector<int> h\_vec(5);

h\_vec[0] = 5;

h\_vec[1] = 1;

h\_vec[2] = 7;

h\_vec[3] = 3;

h\_vec[4] = 2;

// Transfer to device

thrust::device\_vector<int> d\_vec = h\_vec;

// Sort on GPU

thrust::sort(d\_vec.begin(), d\_vec.end());

// Copy back to host

thrust::copy(d\_vec.begin(), d\_vec.end(), h\_vec.begin());

std::cout << "Sorted values:\n";

for (int i = 0; i < h\_vec.size(); i++)

std::cout << h\_vec[i] << " ";

std::cout << std::endl;

return 0;

}

**Compile and Run:**

nvcc -o thrust\_sort thrust\_sort.cu

./thrust\_sort

**3. Summary of CUDA Libraries Demonstrated:**

| **Library** | **Purpose** |
| --- | --- |
| cuBLAS | Dense linear algebra |
| cuFFT | Fast Fourier Transform |
| cuRAND | Random number generation |
| Thrust | High-level vector operations |

**4. Conclusion:**

This lab covered how to:

* Use cuRAND to generate random numbers on the GPU
* Use Thrust for STL-like operations such as sorting
* Compile and execute programs using CUDA auxiliary libraries