

SANSKAAR PATNI  
180905134 CSE C-23  
PP LAB 7  
Programs on Strings in CUDA

1. Write a program in CUDA to count the number of times a given word is repeated in a sentence. (Use Atomic function).

Code:

```
%%cu
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

__global__ void CUDACount(char* A, char *B, int *word_ptr, int
*d_count, int words) {
    int id = threadIdx.x;
    if(id < words)
    {
        int start = word_ptr[id];
        //exclude space
        int end = word_ptr[id+1]-1;
        int j = 0;
        for(int i = start; i < end; i++) {
            if(A[i] != B[j]) {
                break;
            }
            j++;
        }
        if(j == end - start)
        {
            atomicAdd(d_count, 1);
        }
    }
}

int main() {
    char A[] = "mon day tues day wednesday";
    //B should not match with wednesday since day is substring and not a
    separate word
    char B[] = "day";
    //word_ptr concept similar to row_ptr in sparse matrix
    startIdx (includes space)
    int word_ptr[6] = {0, 4, 8, 13, 17, 25};
    char *d_A, *d_B;
    int *d_word_ptr;
    int count = 0;
```

```

int *d_count;
int nofOfWords=5;

cudaMalloc((void **)(&d_word_ptr), 6*sizeof(int));
cudaMalloc((void**) &d_A, strlen(A)*sizeof(char));
cudaMalloc((void**) &d_B, strlen(B)*sizeof(char));
cudaMalloc((void **) &d_count, sizeof(int));

cudaMemcpy(d_word_ptr, word_ptr, 6*sizeof(int), cudaMemcpyHostToDevice);
cudaMemcpy(d_A, A, strlen(A)*sizeof(char), cudaMemcpyHostToDevice);
cudaMemcpy(d_B, B, strlen(B)*sizeof(char), cudaMemcpyHostToDevice);
cudaMemcpy(d_count, &count, sizeof(int), cudaMemcpyHostToDevice);

CUDACount<<<1, nofOfWords>>>(d_A, d_B, d_word_ptr, d_count, nofOfWords);

cudaMemcpy(&count, d_count, sizeof(int), cudaMemcpyDeviceToHost);
printf("Input sentence: %s\n", A);
printf("Total occurrences of word: %s\n%d\n", B, count);

cudaFree(d_A);
cudaFree(d_B);
cudaFree(d_word_ptr);
cudaFree(d_count);
return 0;
}

```

Screenshot:

```

55  cudaFree(d_word_ptr);
56  cudaFree(d_count);
57  return 0;
58 }

```

Input sentence: mon day tues day wednesday  
Total occurrences of word: day  
2

```

1 %%cu
2 #include <stdio.h>

```

2. Write a CUDA program that reads a string **Sin** and produces the string **Sout** as follows:

Input string **Sin**: PCAP Output string **Sout**: PCAPPCAPCP

**Note: Each thread copies required number of characters from Sin in Sout.**

Code:

```

%%cu
#include <stdio.h>
#include <stdlib.h>

```

```

#include <string.h>
__global__ void convertString(char *S_in, char *S_out, int len) {
    int id=threadIdx.x;
    int startIndex=id*len - (id*(id-1)/2);
    int i=0;
    for(i=0;i<len-id;i++){
        S_out[i+startIndex]=S_in[i];
    }
}

int main() {
    char Sin[]="PCAP";
    char Sout[]="abcdefghij";
    char *d_Sin, *d_Sout;
    int length=strlen(Sin);

    int inArray_size=length*sizeof(char);
    int outArray_size=(length*(length+1))/2*sizeof(char);

    cudaMalloc((void **)&d_Sin, inArray_size);
    cudaMalloc((void **)&d_Sout,outArray_size);

    cudaMemcpy(d_Sin, &Sin, inArray_size, cudaMemcpyHostToDevice);

    convertString<<<1,length>>>(d_Sin, d_Sout, length);
    cudaError err = cudaMemcpy(&Sout, d_Sout,
outArray_size,cudaMemcpyDeviceToHost);
    if(err!=cudaSuccess) {
        printf("CUDA error copying to Host: %s\n",
cudaGetErrorString(err));
    }
    printf("2. String Convert\n");
    printf("Input string: %s\n",Sin);
    printf("Resultant String Sout:\n");
    for(int k=0;k<length*(length+1)/2;k++){
        printf("%c",Sout[k]);
    }
    // Cleanup
    cudaFree(d_Sin);
    cudaFree(d_Sout);
    return 0;
}

```

Screenshot:

```
46 }  
  
2. String Convert  
Input string: PCAP  
Resultant String Sout:  
PCAPPCAPCP  
  
1 | %cu
```