

- (1) I can allocate memory using thrust template container like what we do in C++ STL. The following example is to allocate 100 int array in memory using container in device and host.

```
thrust::host_vector<int> host_vector_container(100);
thrust::device_vector<int> device_vector_container(100);
```

Of course, we can allocate the memory manually using thrust; however, we have responsibility to deallocate with device free. The following is short example.

```
thrust::device_ptr<int> device_vector_array = thrust::device_malloc<int>(100);
thrust::device_free(host_vector_array);
```

One useful feature is that we can cast pointer of template container to raw pointer which will be very useful. Since the container is host code, which can not be called in device kernel function, the only way to pass the array directly into device kernel is the raw pointer.

```
thrust::device_ptr<int> host_vector_array
    = thrust::raw_pointer_cast(&device_vector_container[0]);
```

- (2) I have the images in the next pages. There are two images, and the one on the left is the frequency table for cipher_text.txt encrypted by 500 length key. You can see that the distribution is almost uniform across all the letters and digraphs.

However, the one on the right is generated by using plain_text.txt. You can see that e is the most likely letter, and for the bigraphs, there are several recognized patterns.

- (3) I used transform, transform_iterator, remove_copy_if, counting_iterator, sort, upper_bound, sort_by_key, reduce, and inner_product.
- (4) Since the code can not be run with -G flag, here, I mainly compared the performance between -O3 and -g flags. As the following output, the release version with -O3 version is 35% faster than the debugging version.

```
[dbtsai@node006 programming]$ make
nvcc -o create_cipher create_cipher.cu -O3 -arch=sm_20
nvcc -o solve_cipher solve_cipher.cu -O3 -arch=sm_20
[dbtsai@node006 programming]$ time for i in {1..10}; do ./solve_cipher cipher_text.txt > /dev/null; done
real    0m17.005s
user    0m3.460s
sys     0m13.115s
```

```
[dbtsai@node006 programming]$ make clean && make DEBUG=1
rm create_cipher solve_cipher
nvcc -o create_cipher create_cipher.cu -arch=sm_20 -g -G
nvcc -o solve_cipher solve_cipher.cu -arch=sm_20 -g -G
[dbtsai@node001 programming]$ time for i in {1..10}; do ./solve_cipher cipher_text.txt > /dev/null; done

real    0m27.812s
user    0m4.970s
sys     0m21.568s
```

- (5) By adding the text repeated several times over to the plain text, for example, convert the string “The quick brown fox jumps over the lazy dog” into “thequickbrownfoxjumpsoverthelazydogthequickbrownfoxjumpsoverthelazydogthequickbrownfoxjumpsoverthelazydog”, there will be two periodicity involved, which will change the behavior of ioc, and it will cause the solver failed.

```
Terminal - dbtsai@node006:~/2012-04_stanford_cme213/hw/hw3/programmirmir - + x
File Edit View Terminal Go Help
[dbtsai@node006 programming]$ ./solve_cipher cipher_text.txt
Text length: 967673

a: 0.0364596
b: 0.036068
c: 0.0375478
d: 0.038327
e: 0.0394059
f: 0.0384944
g: 0.0360804
h: 0.038668
i: 0.0401468
j: 0.041203
k: 0.0374641
l: 0.0389822
m: 0.0396425
n: 0.0384004
o: 0.0357187
p: 0.037431
q: 0.039471
r: 0.0377876
s: 0.0389119
t: 0.0386122
u: 0.040058
v: 0.0373308
w: 0.0381286
x: 0.0397386
y: 0.042133
z: 0.0377886

Sum of histogram: 1

wy: 0.00205958
yv: 0.00190354
yl: 0.0018839
li: 0.0018777
yq: 0.00183843
ye: 0.00180846
jm: 0.00179813
is: 0.00178573
wx: 0.00178263
ik: 0.00177436
qs: 0.00177126
sy: 0.00176919
iy: 0.00176403
jj: 0.00176403
uj: 0.00175886
xm: 0.00175679
hy: 0.00175576
jf: 0.00175266
ya: 0.00174956
dy: 0.00174853

keyLength: 500
Key: bdqyzfolhudpqwqxztzimryzfyxizljolvspqdmqwnvcfgvdfqrqujmlyzq
kzbcxpzcwezizglwlrzufysbowuyufjpfjslhnlfphuqwgmejsyvfihnhhzityxnc

Terminal - dbtsai@node006:~/2012-04_stanford_cme213/hw/hw3/programmirmir - + x
File Edit View Terminal Go Help
[dbtsai@node006 programming]$ ./solve_cipher plain_text.txt
Text length: 967673

a: 0.0815906
b: 0.0177384
c: 0.0238924
d: 0.0400621
e: 0.12294
f: 0.0219361
g: 0.0219485
h: 0.0655614
i: 0.0688073
j: 0.00121529
k: 0.00847084
l: 0.0447434
m: 0.0244432
n: 0.0688952
o: 0.0730598
p: 0.0183027
q: 0.00161935
r: 0.0551209
s: 0.0671818
t: 0.0925416
u: 0.0281262
v: 0.00901234
w: 0.0232517
x: 0.00109748
y: 0.0177839
z: 0.000657247

Sum of histogram: 1

th: 0.0344156
he: 0.0277635
in: 0.0207798
er: 0.0166709
an: 0.0159806
es: 0.0139159
ha: 0.0131894
st: 0.013117
re: 0.0126375
nd: 0.0118387
at: 0.0111949
ed: 0.0110089
ng: 0.0106265
ea: 0.0106193
hi: 0.010517
nt: 0.0103599
en: 0.0103547
on: 0.0100199
to: 0.00954456
is: 0.0092769
Unusual pattern in text!
[dbtsai@node006 programming]$
[dbtsai@node006 programming]$
[dbtsai@node006 programming]$
[dbtsai@node006 programming]$
```