



A buggy CUDA programs

```

// Kernel
__global__ void count(int *d_countervar){
    d_countervar[0]++;
    __syncthreads();
    d_countervar[0]++;
}
  
```

example of PTX

```

ld.param.u64 %rd2, [_Z5countPi_param_0];
cvta.to.global.u64 %rd1, %rd2;
ld.global.u32 %r1, [%rd1];
add.s32 %r2, %r1, 1;
st.global.u32 [%rd1], %r2;
bar.sync 0;
ld.global.u32 %r3, [%rd1];
add.s32 %r4, %r3, 1;
st.global.u32 [%rd1], %r4;
ret;
} 
  
```

Example of SASS

```

code for sm_35
Function : _Z5countPi
.headerflags @EF_CUDA_SM35 EF_CUDA_PTX_SM(EF_CUDA_SM35)
/* 0x08b8b8a0fc010c0 */
/*0008*/
MOV R1, c[0x0][0x44]; /* 0x64c03c00089c0006 */
/*0010*/
MOV R2, c[0x0][0x140]; /* 0x64c03c00281c000a */
/*0018*/
MOV R3, c[0x0][0x144]; /* 0x64c03c00289c000e */
/*0020*/
LD.E R0, [R2]; /* 0xc4800000001c0800 */
/*0028*/
IADD32I R4, R0, 0x1; /* 0x40000000009c0011 */
/*0030*/
ST.E [R2], R4; /* 0xe4800000001c0810 */
/*0038*/
BAR.SYNC 0x0; /* 0x8540dc00001c0002 */
/*0048*/
LD.E R0, [R2]; /* 0xc4800000001c0800 */
/*0050*/
IADD32I R5, R0, 0x1; /* 0x40000000009c0015 */
/*0058*/
ST.E [R2], R5; /* 0xe4800000001c0814 */
/*0060*/
EXIT; /* 0x18000000001c003c */
/*0068*/
BRA 0x68; /* 0x12007ffffc1c003c */
/*0070*/
NOP; /* 0x85800000001c3c02 */
/*0078*/
NOP; /* 0x85800000001c3c02 */
.....
  
```

Result of DRACC

Friday, May 15, 2020 3:08 PM

- DRACC 001: There are 101 potential data races in 1th kernel lunches
0 of them are intra block shared memory data races in this kernel lunches
100 of them are intra block global memory data races in this kernel lunches
1 of them are inter block global memory data races in this kernel lunches
- DRACC 002: There are 1 potential data races in 1th kernel lunches
0 of them are intra block shared memory data races in this kernel lunches
0 of them are intra block global memory data races in this kernel lunches
1 of them are inter block global memory data races in this kernel lunches
- DRACC 003: There are 1 potential data races in 1th kernel lunches
0 of them are intra block shared memory data races in this kernel lunches
1 of them are intra block global memory data races in this kernel lunches
0 of them are inter block global memory data races in this kernel lunches
- DRACC 004: There are 51198 potential data races in 1th kernel lunches
0 of them are intra block shared memory data races in this kernel lunches
51099 of them are intra block global memory data races in this kernel lunches
99 of them are inter block global memory data races in this kernel lunches
- DRACC 005: There are 99 potential data races in 1th kernel lunches
0 of them are intra block shared memory data races in this kernel lunches
0 of them are intra block global memory data races in this kernel lunches
99 of them are inter block global memory data races in this kernel lunches
- DRACC 006: There are 511 potential data races in 1th kernel lunches
0 of them are intra block shared memory data races in this kernel lunches
511 of them are intra block global memory data races in this kernel lunches
0 of them are inter block global memory data races in this kernel lunches
- DRACC 007: Atomic operations with Recursion Kernel, no data race after add a base case to the recursion kernel.
no data races found in 1th kernel lunches.
- DRACC 027: Synchronized version of DRACC 001, no data race
no data races found in 1th kernel lunches.

DRACC 027: Synchronized version of DRACC 001, no data race

no data races found in 1th kernel lunches.

DRACC 027_injected: I injected data race on DRACC 027 between synchronization operations to test if SRFs are recorded correctly.

There are 215 potential data races in 1th kernel lunches
0 of them are intra block shared memory data races in this kernel lunches
214 of them are intra block global memory data races in this kernel lunches
1 of them are inter block global memory data races in this kernel lunches

DRACC 028: Atomic store, non-atomic load, no data race

no data races found in 1th kernel lunches.

Result of Rodinia 3.1

Friday, May 15, 2020 3:59 PM

b+tree: server crashed

heartwall: server crashed

dwt2d: server crashed

hybirdsort: error in the source code, CUDA error at main.cu:222
have no idea how to fix it

bfs: inter block global memory data race in several kernel lunches
BARRACUDA paper report it is data race free.

```
There are 1 potential data races in 14th kernel lunches
0 of them are intra block shared memory data races in this kernel lunches
0 of them are intra block global memory data races in this kernel lunches
1 of them are inter block global memory data races in this kernel lunches
no data races found in 15th kernel lunches.
no data races found in 16th kernel lunches.
```

The data race reported in my detector is happened on

```
*g_over=true
```

it seems it is a flag and are wrote by several threads without synchronization.

A benign data race?

```
1 __global__ void
2 Kernel2( bool* g_graph_mask, bool *g_updating_graph_mask, bool* g_graph_visited, bool *g_over, int no_of_nodes)
3 {
4     int tid = blockIdx.x*MAX_THREADS_PER_BLOCK + threadIdx.x;
5     if( tid<no_of_nodes && g_updating_graph_mask[tid])
6     {
7
8         g_graph_mask[tid]=true;
9         g_graph_visited[tid]=true;
10        *g_over=true;
11        g_updating_graph_mask[tid]=false;
12    }
13 }
14 }
```

gaussian: BARRACUDA paper report it is data race free.

```
no data races found in 1th kernel lunches.
no data races found in 2th kernel lunches.
no data races found in 3th kernel lunches.
no data races found in 4th kernel lunches.
no data races found in 5th kernel lunches.
no data races found in 6th kernel lunches.
vincent@efeslab-gpu-nn:/CUDA_Data_Race_Detect
```

hotspot: BARRACUDA paper report it is data race free.

It is synchronized by __syncthreads()

A data race will be reported if __syncthreads() is removed

```
no data races found in 1th kernel lunches.
```

kmeans: BARRACUDA paper report it is data race free.

```
no data races found in 1th kernel lunches.  
no data races found in 2th kernel lunches.  
no data races found in 3th kernel lunches.
```

lud: BARRACUDA paper report it is data race free.

```
no data races found in 1th kernel lunches.  
no data races found in 2th kernel lunches.  
no data races found in 3th kernel lunches.  
no data races found in 4th kernel lunches.  
no data races found in 5th kernel lunches.  
no data races found in 6th kernel lunches.  
no data races found in 7th kernel lunches.  
no data races found in 8th kernel lunches.  
no data races found in 9th kernel lunches.  
no data races found in 10th kernel lunches.  
no data races found in 11th kernel lunches.  
no data races found in 12th kernel lunches.  
no data races found in 13th kernel lunches.  
no data races found in 14th kernel lunches.  
no data races found in 15th kernel lunches.  
no data races found in 16th kernel lunches.  
no data races found in 17th kernel lunches.  
no data races found in 18th kernel lunches.  
no data races found in 19th kernel lunches.  
no data races found in 20th kernel lunches.  
no data races found in 21th kernel lunches.  
no data races found in 22th kernel lunches.
```

Overhead is 170x

```
real    2m13.611s  
user    1m55.649s  
sys     0m17.316s  
yinengy@efeslab-gpu-rr:~/  
WG size of kernel = 32 X  
Generate input matrix int  
Creating matrix internal  
Before LUD  
Time consumed(ms): 1.7006  
After LUD  
>>>Verify<<<  
  
real    0m0.746s  
user    0m0.023s  
sys     0m0.606s
```

Problems

Friday, May 15, 2020 3:20 PM

1. (?) Double counts on intra block global memory data race and inter block global memory data race (because they are checked separately)

For example, a program which has all the threads write to a single counter. Will be reported as having two kind of data races at the same time.

2. Cannot tell the position of data races in the source code. Number of data races are counted related to memory address rather than instruction.

Even though there is only 1 instruction that has data race, if the memory address changed, the data race will be reported separately.

3. Crushed on large input (e.g. Rodinia 3.1 b+tree, dwt2d, heartwall).

4. Is memory fence a synchronization operations that can produce SFRs?

5. Fine granularity synchronization operations like CUDA Cooperative groups are not supported (introduced in CUDA 9).

However, Rodinia doesn't use any of them.

6. SASS is hard to understand!