A deviation of CURAND: standard pseudorandom number generator in CUDA for GPGPU

Mutsuo Saito[1], Makoto Matsumoto[2]

[1]Hiroshima University, [2]University of Tokyo

February 13, 2012

This study is granted in part by JSPS Grant-In-Aid #23244002, #21654004, #21654017.

Introduction

CUDA, CURAND, `xorwow`

- CUDA is a developing environment for General Purpose computation by Graphic Processing Units (GPGPU).
- In 2010 August, CUDA released CURAND, a library for pseudorandom number generation.
- There, `xorwow` generator (xor-shift added with Weyl sequence, introduced by Marsaglia in 2003) was selected as the standard.

Deviation of `xorwow`

- It was reported (in the web) that `xorwow` is rejected by one of the tests in BigCrush test-suite in TESTU01 (L'Ecuyer-Simard).
- We analyze some weakness of `xorwow`.
- The six dimensional distribution has an observable deviation.
- Its difference sequence is more clearly rejected by BigCrush.

`xorwow` generator: `xorshift`-part

`xorwow` = `xorshift`+Weyl generator.

- `xorshift` generator:

  Let $\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_n, \ldots$ be a sequence of 32-bit integers.

  `xorshift` generator (by Marsaglia) generate such a sequence by the following recursion formula:

$$\begin{aligned} \mathbf{x}_{n+5} &= \mathbf{x}_{n+4} \oplus (\mathbf{x}_{n+4} << 4) \oplus \\ & \quad \mathbf{x}_n \oplus (\mathbf{x}_n << 1) \oplus ((\mathbf{x}_n >> 2) << 1) \oplus (\mathbf{x}_n >> 2). \end{aligned}$$

  Here,

  - $\oplus$ denotes bit-wise XOR,
  - $(\mathbf{x} << m)$ denotes m-bit shift-left,
  - $(\mathbf{x} >> m)$ denotes m-bit shift-right.

  $\mathbb{F}_2$-linear generator, period $2^{5\times 32} - 1 = 2^{160} - 1$.

xorwow generator: Weyl part

- Weyl generator: (Marsaglia)
  Generate a sequence of 32-bit integers $\mathbf{y}_0, \mathbf{y}_1, \ldots, \mathbf{y}_n, \ldots$ by

  $$\mathbf{y}_{n+1} = \mathbf{y}_n + 362437 \bmod 2^{32},$$

  period 2^{32}.

The output sequence $\mathbf{z}_0, \mathbf{z}_1, \ldots$ of xorwow is the sum of the two sequences:

$$\mathbf{z}_n := \mathbf{x}_n + \mathbf{y}_n \bmod 2^{32}.$$

Its period is $(2^{160} - 1)2^{32}$.

Three tests in TESTU01 reject xorwow

- TESTU01 statistical test suit (Simard-L'Ecuyer) clearly reject both xorshift and Weyl generators.
- Among the 106 tests in BigCrush, the following three tests systematically reject xorwow:
  - Test 7: Collision Over test on the 7-dimensional distribution (each axis is partitioned into 64 equal length interval, hence 7-dimensional unit cube is partitioned into 64^7 small cells, and count the number of collisions among 2×10^7 points generated by overlapping 7 tuples from the generator; 30 times iterated).
    p-value: around $10^{-4} \sim 10^{-6}$
  - Test 27: Simplified Poker Test for the five least significant bits (on the 8 dimensional distribution).
    p-value: around $10^{-16} \sim 10^{-300}$
  - Test 81: Linear Complexity Test for the three least significant bits.
    p-value: around $1 - 10^{-15}$

Three tests in TESTU01 reject xorwow

These results show that some flaw is hidden in xorwow generator. In particular, rejection by Test 7 is serious for MonteCarlo simulations, since it is about the six most significant bits (the rest two are on the least significant bits, and the latter one is on the $\mathbb{F}_2$-linearity, which seems not very significant for Usual MonteCarlo).

Analysis of defects of xorwow

By mathematical analysis, we found that xorwow has a significant deviation on 6-dimensional distribution of the most significant 5 bits.

- Recall that xorwow sequence is $\mathbf{z}_n = \mathbf{x}_n + \mathbf{y}_n \bmod 2^{32}$, where $\mathbf{x}_n$ is from xorshift and $\mathbf{y}_n$ is from Weyl generator.

Analysis of defects of xorwow (continued)

- xorshift is generated by

$$\begin{aligned}\mathbf{x}_{n+5} &= \mathbf{x}_{n+4} \oplus (\mathbf{x}_{n+4} << 4) \oplus \\ &\quad \mathbf{x}_n \oplus (\mathbf{x}_n << 1) \oplus ((\mathbf{x}_n >> 2) << 1) \oplus (\mathbf{x}_n >> 2).\end{aligned}$$

- Let $\mathbf{x}_n(i)$ denote the i-th bit from the MSB. One sees that there is a simple relation among seven bits in every consecutive 6-tuples for every i (except $i = 32$) :

$$\begin{aligned}\mathbf{x}_{n+5}(i) &= \mathbf{x}_{n+4}(i) \oplus \mathbf{x}_{n+4}(i+4) \\ &\quad \mathbf{x}_n(i-2) \oplus \mathbf{x}_n(i-1) \oplus \mathbf{x}_n(i) \oplus \mathbf{x}_n(i+1).\end{aligned}$$

  In particular, the 5 MSBs have a simpler relation

$$\begin{aligned}\mathbf{x}_{n+5}(1) &= \mathbf{x}_{n+4}(1) \oplus \mathbf{x}_{n+4}(5) \oplus \\ &\quad \mathbf{x}_n(1) \oplus \mathbf{x}_n(2).\end{aligned}$$

Analysis of defects of xorwow (continued 2)

- Thus, 6-dimensional distribution of the 5 MSBs of $(\mathbf{x}_n)$ is rather deviated.

$$\begin{aligned} \mathbf{x}_{n+5}(1) &= \mathbf{x}_{n+4}(1) \oplus \mathbf{x}_{n+4}(5) \oplus \\ & \quad \mathbf{x}_n(1) \oplus \mathbf{x}_n(2). \end{aligned}$$

  E.g. if $\mathbf{x}_n < 2^{32-2}$ and $\mathbf{x}_{n+4} < 2^{32-5}$ hold, then $\mathbf{x}_{n+5} < 2^{32-1}$. When the 32-bit integers are normalized into [0,1]-interval, $\mathbf{x}_n < 1/4$ and $\mathbf{x}_{n+4} < 1/32$ imply $\mathbf{x}_{n+5} < 1/2$.

- The choice of 362437 in the Weyl generator

$$\mathbf{y}_{n+1} = \mathbf{y}_n + 362437 \bmod 2^{32},$$

  is too small compared to 2^{32}. Namely, the most significant 6 bits in $\mathbf{y}_n$ change seldomly when n is incremented. The change occurs once in every $2^{32-6}/362437 = 185.16$ times generation of y_n.

Analysis of defects of xorwow (continued 3)

- Thus, the value of the following formula from the output $\mathbf{z}_n$ of xorwow

$$\mathbf{z}_{n+5}(1) \oplus \mathbf{z}_{n+4}(1) \oplus \mathbf{z}_{n+4}(5) \oplus \mathbf{z}_n(1) \oplus \mathbf{z}_n(2) \quad (***)$$

  is 0 when $\mathbf{y}_n, \ldots, \mathbf{y}_{n+5}$ are smaller than 2^{32-6} (Since then adding $\mathbf{y}_n$'s does not change the 5 MSBs).
  More generally, if $\mathbf{y}_n, \ldots, \mathbf{y}_{n+5}$ share the same 6 MSBs of type ?00??0, then the value of (***) depends only the the pattern ?00??0. (Since such 32-bit integers do not cause a carry to the 1st, 2nd and 5th.) Thus, the value of (***) is 0 for a while (or 1 for a while).

The xor (***) of the five bits for 1000 generations

012345678901234567890123456789012345678901234567890
000000000000000000000000000000000000000000000000000
000000000000000100000000000010001011000000010110000
001100000011000000101100000001010010010100101000011
001100111110000100011010110000000110000001101111000
111001111000111010111110110101000111011101001011011
110100110101011111011011111111111100111110111101000
111110111110111111111010110101101111111000111111111
111111100111111111100011111111111111111111111111001
111111111011111011100000111101010111000011110111110
111001011011111111110011110101001001001000000000010
110110011100110101011011010111010101101011101100100
010111010100101000100100101001011001100110100001111
001110000010000101000001011100000000100011010001000
011100110100000001110000000111111100101000010000111
010001001010000000011000000011000000000000101011100
000011000011111100011100001000000100000010000100000

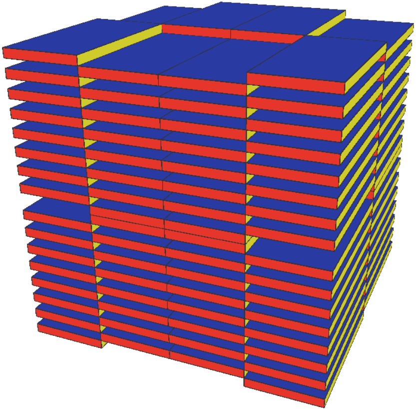
The xor (***) of the five bits for 1000 generations

012345678901234567890123456789012345678901234567890
0100000101110000000111000010100111111001010011110001
000011010111001101100100111010110001010101111111000
101010010111101010001110110110010101110101110111111
110100100011101000011111110100110101011110111111010
011101111110110111111111111111111101111111111111111
011111111111111111111111111111111111111111111111111
110111111011111110111111101111111111101111101110110
111110111111111111101100111101111111110001000110111
110111111101111000110111100010011011110011111100000
110110111101111000101100110100000100010000110110000
0101010001111011100010011000000001001000010010000110
001101001010000110100000000000111000000100000001010000
000011001000000000101100110001011000000100000000110
000010000001100100110000000100010001000000000010000000
000001100010100000001010000000101010001000100000010011
011001001000100100101000100000101011001000101001001

Toy experiments: volume of a part W

We identify 32-bit integers with 2^{32} intervals in $[0,1]$. Define $W \subset [0,1]^6$ as the set of $(z_0, z_1, \ldots, z_5)$ such that
$\mathbf{z}_5(1) \oplus \mathbf{z}_4(1) \oplus \mathbf{z}_4(5) \oplus \mathbf{z}_0(1) \oplus \mathbf{z}_0(2) = 0$ holds. (W for Walsh.)
6-tuples from $(\mathbf{x}_n)$ always fall in W.
6-tuples from $(\mathbf{z}_n)$ falls in $W \Leftrightarrow$ the value of (***)$= 0$.

Toy experiments: volume of a part W



This is the projection of W to the three dimensional cube by $(\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3, \mathbf{z}_4, \mathbf{z}_5, \mathbf{z}_6) \mapsto (\mathbf{z}_1, \mathbf{z}_5, \mathbf{z}_6)$. W is the inverse image. The above picture denotes the region where $\mathbf{z}_6(1) \oplus \mathbf{z}_5(1) \oplus \mathbf{z}_5(5) \oplus \mathbf{z}_1(1) \oplus \mathbf{z}_1(2) = 0$. Its volume is the half of the volume of the unit cube).

Toy experiments: volume of a part W

Use xorwow to estimate the volume of W, by generating 100 points in $[0,1]^6$ (use non-overlapped 6-tuples).

hit	dev	p-value	hit	dev	p-value
60	2.0	0.999968328758167	41	-1.8	0.000159108590158
51	0.2	0.655421741610324	51	0.2	0.655421741610324
45	-1.0	0.022750131948179	47	-0.6	0.115069670221708
44	-1.2	0.008197535924596	55	1.0	0.977249868051821
57	1.4	0.997444869669572	47	-0.6	0.115069670221708
44	-1.2	0.008197535924596	47	-0.6	0.115069670221708
52	0.4	0.788144601416603	52	0.4	0.788144601416603
58	1.6	0.999312862062084	36	-2.8	0.000000010717590
57	1.4	0.997444869669572	45	-1.0	0.022750131948179
49	-0.2	0.344578258389676	43	-1.4	0.002555130330428
58	1.6	0.999312862062084	52	0.4	0.788144601416603
42	-1.6	0.000687137937916	52	0.4	0.788144601416603
55	1.0	0.977249868051821	46	-0.8	0.054799291699558

Toy experiments: volume of a part W (continued)

hit	dev	p-value	hit	dev	p-value
54	0.8	0.945200708300442	50	0.0	0.500000000000000
46	-0.8	0.054799291699558	63	2.6	0.999999900355737
44	-1.2	0.008197535924596	46	-0.8	0.054799291699558
55	1.0	0.977249868051821	49	-0.2	0.344578258389676
59	1.8	0.999840891409842	56	1.2	0.991802464075404
53	0.6	0.884930329778292	47	-0.6	0.115069670221708
45	-1.0	0.022750131948179	31	-3.8	0.000000000000015
56	1.2	0.991802464075404	62	2.4	0.999999206671848
49	-0.2	0.344578258389676	62	2.4	0.999999206671848
49	-0.2	0.344578258389676	55	1.0	0.977249868051821
52	0.4	0.788144601416603	46	-0.8	0.054799291699558
44	-1.2	0.008197535924596	57	1.4	0.997444869669572
58	1.6	0.999312862062084	45	-1.0	0.022750131948179
52	0.4	0.788144601416603	55	1.0	0.977249868051821
46	-0.8	0.054799291699558	57	1.4	0.997444869669572

CollisionOver test revisited

In BigCrush, 7-dimensional Collision Over Test deals with the 6 MSBs. When we test the 5 MSBs in the same manner, then the p-values become far smaller: $< 10^{-60}$ (too many collisions).

Another defect: difference sequence

Let (z_n) be the output of xorwow. Define its difference sequence (d_n) by

$$d_n := z_{n+1} - z_n \bmod 2^{32}.$$

The results of 106 tests in BigCrush on d_n (eps means a value $< 10^{-300}$):

Test	p-value
7 CollisionOver, t = 7	6.0e-74
8 CollisionOver, t = 7	1.6e-45
10 CollisionOver, t = 14	6.1e-36
36 Gap, r = 0	1.7e-13
38 Run, r = 0	1.7e-4
75 RandomWalk1 H (L=50, r=25)	eps
75 RandomWalk1 M (L=50, r=25)	eps
96 HammingIndep, L=30, r=27	2.4e-157
101 Run of bits, r = 0	2.6e-5
102 Run of bits, r = 27	7.8e-16

A reason why $d_n := z_{n+1} - z_n$ fails clearer

- $z_n := x_n + y_n \bmod 2^{32}$.
-

$$\begin{aligned} d_n &= (x_{n+1} + y_{n+1}) - (x_n + y_n) \\ &= (x_{n+1} - x_n) + (y_{n+1} - y_n) \bmod 2^{32} \\ &= x_{n+1} - x_n + 362437 \bmod 2^{32}. \end{aligned}$$

  y_n elliminated.
- As we saw, the output (x_n) of xorshift has obvious relations among a few number of bits in consecutive 6 tuples, and d_n inherits the deviation.

Conclusion

- xorwow is not suitable for serious MonteCarlo. (Note: Panneton-L'Ecuyer analyzed xorshift and warned on its deviation in 2004).
- A choice of small value 362437 in the Weyl generator caused serious deviation in 6-dimensional distribution of the 5 MSBs.
- Deviation persist for the LSBs, when 362437 is repaced to a large number: We did not mention, but LSBs have more serious deviations. Note that k LSBs of Weyl generator has period $\leq 2^k$, for any choice of d in $y_{n+1} := y_n + d$.
- Anyway, ad-hoc modification of xorwow seems potentially dangerous. Why not use generators having assurance on high dimensional equidistribution property?

Conclusion-Advertise

- We have Mersenne Twister for GPGPU (MTGP, 2010) with period $2^{11213}-1$ and 175-dimensional equidistribution property, passing BigCrush (except those on $\mathbb{F}_2$-linearity). This MTGP and Multiplicative Recursive Generator were included in CURAND (Jan. 2012) as other choices (than the STANDARD xorwow).
- We developped and released "tiny Mersenne Twister" (tinyMT, 2011) with period $2^{127}-1$ whose MSBs and LSBs have high dimensional equidistribution property, passing all tests in BigCrush. (Some non-linearity introduced.)
- Many distinct parameters for these generators, and Dynamic Creators to generate such parameters are also released.
- Downloadable from "Mersenne Twister Homepage".

Thank you for listening.