SQUAREMIX: A Faster Pseudorandom Number Generator for Dynamic-Multithreading Platforms

Robert Ritchie and Khodakhast Bibak

Department of Computer Science and Software Engineering
Miami University
Oxford, Ohio, 45056, USA

`{ritchirp,bibakk}@miamioh.edu`

Introduction

Many concurrency platforms offer a processor oblivious model of computation, where the scheduler dynamically distributes work across threads. While this is convenient, it introduces non-determinism at runtime, which complicates debugging, since it precludes repeatability. Leiserson *et al.* [PPoPP '12] persuaded Intel to modify its C/C++ compiler, which provided the Cilk Plus concurrency platform, to include a feature called pedigrees, which enables determinism by uniquely identifying strands with low overhead. They used pedigrees to design a deterministic parallel random number generator (DPRNG), called DOTMIX, which hashes a pedigree using a specific compression function, and then runs the output of the compression function through a mixing function based on the RC6 block cipher.

DOTMIX

Assume that the width of a machine word on our architecture is w bits. Pick a prime $p < m = 2^w$. For an arbitrary instruction x, assume that its *spawn depth* (the length of its path on the invocation tree) is bounded, i.e., $d(x) \leq D$ for some $D \in \mathbb{Z}$. Then the pedigree for an instruction, $J(x)$, can be represented as a vector $J(x) = \langle j_1, j_2, \ldots, j_D \rangle \in \mathbb{Z}_p^D$, where for $i > d(x)$ we have $j_i = 0$.

For a given pedigree $J \in \mathbb{Z}_p^D$, the random number generated by DOTMIX is the output of the function $h : \mathbb{Z}_p^D \to \mathbb{Z}_p$ which is composed of a compression function and a mixing function. The compression function chooses a vector Γ uniformly at random from $\mathbb{Z}_p^D$, computes its dot product with J, and then outputs the result modulo p. More formally:

Let $\Gamma = \langle \gamma_1, \gamma_2, \ldots, \gamma_D \rangle$ be chosen uniformly at random from $\mathbb{Z}_p^D$. Define the compression function $c_\Gamma : \mathbb{Z}_p^D \to \mathbb{Z}_p$ by

$$c_\Gamma(J) := J \cdot \Gamma \pmod{p} = \sum_{i=1}^{D} j_i \gamma_i \pmod{p},$$

where $J = \langle j_1, j_2, \ldots, j_D \rangle \in \mathbb{Z}_p^D$. The DOTMIX compression function family is the set

$$C_{\text{DOTMIX}} := \{c_\Gamma : \Gamma \in \mathbb{Z}_p^D\}.$$

Leiserson *et al.* [PPoPP '12] proved that the DOTMIX compression function family C_{DOTMIX} is a universal family of hash functions (in fact one can show that it is even Δ-universal).

Even though the universality of the family C_{DOTMIX} means that the collision probability is minimal, the hash values of two similar pedigrees may still be similar. For this reason, DOTMIX runs the output of the compression function through a mixing function based on the RC6 block cipher defined below. For any w-bit input z, with w even, let $\phi(z)$ be the function which swaps the high and low order $w/2$ bits of z, namely,

$$\phi(z) = \lfloor \tfrac{z}{\sqrt{m}} \rfloor + \sqrt{m}(z \mod \sqrt{m}),$$

and let

$$f(z) = \phi(2z^2 + z) \mod m.$$

Problem

Leiserson *et al.* [PPoPP '12] proposed as "an interesting topic for future research" the use of a faster hash function for DOTMIX. In this paper, we address this question. We propose a new construction and show that it is much faster than DOTMIX, without sacrificing any statistical quality. Like DOTMIX, our scheme (which we call SQUAREMIX) is comprised of two stages, compression and mixing. The mixing function that we use is the same as that of DOTMIX, but for compression we use a faster universal hash function family due to Etzel *et al.* [CRYPTO '99] called Square Hash (constructed for applications in message authentication).

SQUAREMIX

Like DOTMIX, our scheme is comprised of two stages, compression and mixing. Let w be the width of a machine word on our architecture, and pick a prime $p < m = 2^w$. Assume that for an arbitrary instruction x, the spawn depth is bounded by some constant D. For a given pedigree $J = \langle j_1, j_2, \ldots, j_D \rangle \in \mathbb{Z}_p^D$, SQUAREMIX generates a random number by compressing the pedigree, then putting it through a mixing function. The SQUAREMIX compression function s_Γ hashes the pedigree with a vector $\Gamma = \langle \gamma_1, \gamma_2, \ldots, \gamma_D \rangle \in \mathbb{Z}_p^D$ as follows.

Let $\Gamma = \langle \gamma_1, \gamma_2, \ldots, \gamma_D \rangle$ be chosen uniformly at random from $\mathbb{Z}_p^D$. Define the compression function $s_\Gamma : \mathbb{Z}_p^D \to \mathbb{Z}_p$ by

$$s_\Gamma(J) := \sum_{i=1}^{D} (j_i + \gamma_i)^2 \pmod{p},$$

where $J = \langle j_1, j_2, \ldots, j_D \rangle \in \mathbb{Z}_p^D$. The SQUAREMIX compression function family is the set

$$S_{\text{SQUAREMIX}} := \{s_\Gamma : \Gamma \in \mathbb{Z}_p^D\}.$$

We replaced the compression function family C_{DOTMIX} used in DOTMIX with $S_{\text{SQUAREMIX}}$. Since, by the result of Etzel *et al.* [CRYPTO '99], $S_{\text{SQUAREMIX}}$ is a Δ-universal family of hash functions, the statistical quality of the corresponding PRNG will not change.

Since $S_{\text{SQUAREMIX}}$ is universal, there is low probability of collision, but similar pedigrees may yet hash to similar values. For this reason, we run the output of the compression function s_Γ through the RC6-based mixing function.

Main Result (High Level Description)

We show that SQUAREMIX executes significantly faster. Specifically, we prove that SQUAREMIX is roughly twice as fast as DOTMIX.

We also show that it is possible to further improve the speed of the algorithm but potentially at the expense of lowering the statistical quality.

Applications

DOTMIX is a fast and reliable DPRNG which is comparable to the seminal Mersenne Twister (which is the default PRNG for many software systems) in terms of statistical quality. It has found various industry applications. DOTMIX was incorporated into Intel Cilk Plus and both the Intel and GNU C/C++ compilers [Schardl, 2016]. It has also been used in Intel's DPRNG library [Schardl, 2016]. Steele *et al.* [OOPSLA '14], inspired by DOTMIX, designed SPLITMIX which has been included in Java JDK8 as the class `java.util.SplittableRandom`. As SPLITMIX also uses the DOTMIX compression function, it might be possible to also improve the SPLITMIX speed using SQUAREMIX. It would be interesting to investigate the impact of SQUAREMIX in these or other applications.