

When you start talking about numbers as small as 2^{-122} , you have to start looking more closely at the things you thought were zero

devblogs.microsoft.com/oldnewthing/20160114-00

January 14, 2016



Raymond Chen

GUID generation algorithm 4 fills the GUID with 122 random bits. The odds of two GUIDs colliding are therefore one in 2^{122} , which is a phenomenally small number. When you are dealing with rates this low, you have to adjust your frame of reference.

Some people don't like algorithm 4 because the algorithm is not designed to ensure uniqueness. There is a chance, admittedly small, that a collision will occur. On the other hand, algorithm 1, for example, uses time-space coordinates to ensure zero chance of duplication, unless someone screws up the generation process.

Hang on a second. "Unless someone screws up the generation process." Ah, so it's not zero after all, because there is of course a tiny chance that something will get screwed up in the generation process.

According to [this research paper \(pdf\)](#), memory chips suffer 25,000 to 70,000 errors per billion hours per megabit. Let's do some back-of-the-envelope calculations.

$$\begin{array}{rcl} 25,000 \text{ to } 70,000 \text{ errors} & \approx & 2^{15} \text{ errors} & = & 1 \text{ error} \\ \hline 10^9 \text{ hours} \times 2^{20} \text{ bits} & & 2^{30} \text{ hours} \times 2^{20} \text{ bits} & & 2^{35} \text{ hours} \times \text{bits} \end{array}$$

Now let's see how this error rate converts to "corrupted GUIDs".

$$\begin{array}{rcl} 1 \text{ error} & \times & 2^7 \text{ bits} & \times & 1 \text{ hour} \\ \hline 2^{35} \text{ hours} \times \text{bits} & & 1 \text{ GUID} & & 60 \times 60 \times 10^9 \text{ nanoseconds} \end{array}$$

Now, $60 \times 60 = 3600$ is approximately $4096 = 2^{12}$, and 10^9 is approximately 2^{30} .

$$\approx \frac{2^7 \text{ errors}}{2^{35} \times 2^{12} \times 2^{30} \text{ GUID} \times \text{nanosecond}} = \frac{1 \text{ errors}}{2^{70} \text{ GUID} \times \text{nanosecond}}$$

The odds that a GUID will be corrupted in memory in the one nanosecond between the time it is created and the time the value is copied somewhere safe is therefore around one in 2^{70} .

This is $2^{52} \approx 10^{15}$ = a quadrillion* times more likely than a collision between two GUIDs generated by algorithm 4.

And that's just mechanical failures in the RAM chips. There are other ways that the generation process can get screwed up. For example, there could be a failure in the CPU itself, maybe due to overheating or overclocking. Or a bit error on the hard drive. Or a programming error.

When you start talking about numbers as small as 2^{-122} , you have to start looking more closely at all the things that you thought were zero.

* Or "a thousand billion" if you use the European scale.

Bonus chatter: Note that I'm not saying that a collision caused by a random bit flip is more likely than a collision caused by a v4 GUID happening to match some other v4 GUID. I'm saying that it's wrong to say that the v1 GUID generator will never generate a collision, because there's a nonzero probability that the v1 GUID generator will deviate from the algorithm. Sure, that probably may be minuscule, but when you're dealing with numbers like 2^{-122} , these minuscule probabilities become significant.

Raymond Chen

Follow

