The Probability Analysis Behind the Random Tick Update in Minecraft

By: Runaway_Fancy

Contents

1 Write in Front

This article is mainly explain the mechanics of random tick under Minecraft java edition. This small really takes lots of my times as the final exam is in the next week. I would like to use English to describe for my favor. So, it's arduous for Chinese players to read it over. But, actually, I would appreciate if you could find helpful parts for you and your machines.

2 Chunk Tick

2.1 Chunks, Sub-chunks

Before we start, we should know the structure of chunk and sub-chunk. A chunk contains $16 \times 16 \times 256$ blocks starting from layer $y = 0$ to $y = 256$(before 1.17). A sub-chunk is constructed by $16 \times 16 \times 16$ blocks forms a cubic. Each chunk contain 16 sub-chunks.

2.2 Definition

Chunk tick is one type of the cycle of the game. It happens inside a cylinder with horizontal cross section parallel to the horizontal surface with radius $R = 128$ blocks and central chunk ticking as **entity ticking**. **The block inside the ticking chunks are ticked on every game tick**.

2.3 Example of Events Controlled by Chunk Tick update

There are some events will happen if chunks gets ticked[1].

· Mob naturally spawn.

· During a thunderstorm, lightning may strike somewhere in the chunk.

· If in a cold biome, water freezes into ice if possible.

· If snowing, a snow layer is placed if possible(cauldrons can be filled with powder snow, in 1.17).

· A certain number of blocks within the chunk receive random block ticks, that is what we focus on.

3 Random Tick

3.1 Definition

Random tick can be considered as a special case of chunk tick. It happens in every sub-chunks that inside the ticked chunks. The mechanics of this random tick update is(under the rate equal to 3)

Randomly select three blocks in one sub-chunks and give them random tick update

Default value of random tick rate is 3. Players could also use command */gamerule randomTickSpeed <value>* to switch the random tick rate. There is one particular case that even though some blocks are out of the range of the ticking cylinder, it can also receive random tick update.

3.2 Random Tick Dependent Blocks

Most of blocks do not affect by random tick update. But some use it to do something[1].

· Crops may grow or uproot.

· Mushrooms may spread or uproot.

· Vines may spread.

· Fire may burn out or spread.

· Ice and snow layers may melt.

· Leaves may decay.

· Farmland hydration is updated.

· Cacti, sugar cane, kelp, bamboo, chorus flowers and sweet berry bush may grow.

· Grass blocks and mycelium may spread.

· Grass blocks, mycelium, and nylium may decay (if and only if the condition is met).

· Saplings may grow into a tree.

· Lava may set fires nearby.

· Lit redstone ore turns off.

· Turtle eggs crack or hatch.

· Campfire smoke appears.

4 Random Tick Update inside One Sub-chunk

It seems that random tick is a total random thing that we are forceless to control this thing. But we could use statical analysis to predict the tendency of random tick or use probability and expectation value to make our decision.

4.1 The Probability Distribution Model of Random Tick Update

For a random block in one sub-chunk, the probability that one random observed block is ticked is

$$p = \frac{3}{\Omega} \tag{1}$$

Where $\Omega = 16^3$. Assuming the random variable K represents "The number of a random block be ticked". For observing time period t, we have

$$K \sim b(t, p) \tag{2}$$

The probability mass function of K is

$$P(K = k)(t) = C_k^t p^k (1 - p)^{t-k} \tag{3}$$

Where

$$C_k^t = \frac{t!}{k!(t-k)!} \tag{4}$$

The cumulative distribution function is

$$C(t) = \sum_{j}^{t} P(K = k)(j) \tag{5}$$

Or using poisson distribution(Ω is big enough, binomial distribution can approximate to poisson distribution).

4.2 Modification of our Model

We aware that there is period form the last ticking to the end of observation which equivalently is doing nothing. So, to maximize the efficient, we cloud change a little bit on our probability model

$$P(K = k)(t) = C_{k-1}^{t-1} p^k (1 - p)^{t-k} \tag{6}$$

This is known as **negative binomial distribution**. That means we fixed our t^{th} observation whose result is "ticking". After achieving our quota, we stop observing.

5 In Real Application - Modify the Construction of Random Tick Depended Farm

5.1 Condition probability

In real cases, some blocks inside one sub-chunks is not our targets, e.g. a cocoa farm may contain n cocoa-beans in one sub-chunk($n \geq 3$). Assuming random variable X represents the number of cocoa were ticked. $X \sim G(3, \Omega)$, the probability distribution is listed in table 1.

Notation	p_0	p_1	p_2	p_3
X	0	1	2	3
$P(X = x)$	$\frac{C_3^{\Omega-n}}{C)3^{\Omega}}$	$\frac{C_2^{\Omega-n}C_1^n}{C)3^{\Omega}}$	$\frac{C_1^{\Omega-n}C_2^n}{C)3^{\Omega}}$	$\frac{C_3^n}{C)3^{\Omega}}$

Table 1: G.D. for $n \geq 3$

If we observe a random cocoa bean a_i, the conditional probability $P(a_i|X = b)$ means "The probability of ticking a_i under the condition only tick b cocoa beans totally". So we have

$$P(\text{ticking } a_i \text{ and only tick b cocoa beans}) = \frac{b}{n}p_b \quad (b = 0, 1, 2, 3) \tag{7}$$

The total probability that a random cocoa bean is ticked

$$P_\Sigma = 0.2(\frac{1}{n}p_1 + \frac{2}{n}p_2 + \frac{3}{n}p_3) \tag{8}$$

The factor 0.2 is the probability tha the cocoa bean change its level when received one random tick.

5.2 General Formula

The general formula of calculate the productive efficiency is

$$\Gamma(t) = \frac{C(t)}{t} \times N\gamma\epsilon \tag{9}$$

Where

$$C(t) = \sum_{j=m}^{t} P(M = m, j) \tag{10}$$

$$P(M = m, j) = C_{m-1}^{j-1} p_\Sigma^m (1 - p_\Sigma)^{j-m} \tag{11}$$

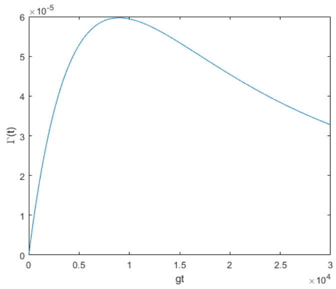
$$P_\Sigma = r(\frac{1}{n}p_1 + \frac{2}{n}p_2 + \frac{3}{n}p_3) \tag{12}$$

For P_i, seen in table 1. r here means the probability that the target block change level after receive one random tick. m means the number of successfully update. M is random variable. N is the number of the block per sub-chunk. γ is drop rate. ϵ is collected rate.

Here we provide a MATLAB script to help players to calculate and find the critical point of $\Gamma(t)$, see figure 1.

```
p = 0.0002;%The probability that the target block successfully get ticked
m = 2;%Integer, how many ticked need to receive per target block
t=m:1:30000;
E=1;
for i = 1:m-1
    E = (t-i).*E;%using E to replace the fraction and factorial in P
end
P = E .*p.^m .* (1-p).^(t-m);%PMF
C = cumsum(P)./t;%cummulate sum CDF
plot(t,C)
```

(a) MATLAB program to calculate the maximum of $\Gamma(t)$



(b) Result of this example

Figure 1: MATLAB program and the result of cocoa bean farm modification

5.3 Other example

The is another mature example which is written by TokiNoBug. Here is the link of that article. Calculated Nether Wart Farm.

References

[1] Wikipedia Minecraft