

排队这件小事

为什么服务器同时只写两本书，第三本老实排队

王建硕

平实随笔 · 由 AI 代理撰写并互相审校

导读

为什么服务器同时只写两本书，第三本老实排队

先问一个问题：三本书摆在面前，一台服务器，是一起写先写完，还是排队写先写完？

直觉说，当然是一起写——三管齐下，谁也不等谁。可直觉在这件事上靠不住。这本小书就讲这一个道理：从我家厨房的三只锅讲起，讲到计算机里「换脑子」的成本，最后落到我们这台写书服务器立下的一条规矩。

道理很小，小到一句话能说完；但它是拿三份半成品的代价换来的。三章，几百字，读完你再决定：下次遇到「为什么不同时开工」，你会怎么回答。

第一章 · 三只锅同时烧

我家厨房有三个灶眼。有年过年，我心血来潮，三只锅同时开火：左边炖肉，中间炒菜，右边煮汤。我信心满满，觉得这样最快——三管齐下嘛。

结果你猜得到。给肉翻面的时候，菜糊了；关火救菜的时候，汤溢了。最后三道菜没一道像样，收拾灶台还多花了半小时。那天我学到一件事：**同时做三件事，常常等于一件都做不好。**

这不是我手艺差，是规律。人的注意力就一份，锅却有三只。你分给每只锅的，不是三分之一的厨艺，而是三分之一再打一个折——因为每次转头，你都得先想起来「刚才那只锅到哪一步了」。这个「想起来」的功夫，看不见，但真金白银地花掉了。

后来我管这个叫「三只锅定律」：事情一多，失败的方式不是变慢，而是**一起垮掉**。炖肉不会因为你在炒菜就温顺地等你，它照糊不误。

这个道理放在厨房里是常识，放到别的地方，人就容易犯糊涂。比如我们这台写书的服务器，就曾经一口气接了三本书。

第二章 · 切换是有代价的

上一章说的「想起来的功夫」，在计算机里有个正式名字，叫上下文切换——大白话说，就是从一个活儿换到另一个活儿时，先把脑子里的东西倒出来、再装回去的那个过程。

关键在于：**切换本身不干活，只花钱**。你放下书 A 去写书 B，先得把 A 写到哪儿、主角叫什么、埋了什么伏笔记在别处；回头写 A 时再一条条装回来。记和装，都是时间，但读者一个字也看不到。

更要命的是，这个开销不随任务变少。三本书并行，每本只分到三分之一的算力，可每本的「记了装、装了记」一分不少。极端情况下，服务器大部分时间都在倒腾脑子，真正写字的时间所剩无几——三本书都写到一半，谁也没熬过谁，最后一起崩掉，前功尽弃。

大白话

大白话：并行不是三份力气同时使，而是三份力气各扣一笔「换脑子税」。税一高，活儿就白干。

所以「同时做」和「做得快」是两回事。那怎么办？答案朴素得让人不好意思：**排队**。

第三章 · 排队不是怠慢，是承诺

现在我们这台写书的服务器立了条规矩：**同时最多写两本，第三本排队。**

听起来像是怠慢——凭什么让第三本等？但换个角度看，排队恰恰是一种承诺。它承诺的是：凡是被接手的书，一定能写完。前两本专心写到结尾，第三本再堂堂正正上场，同样被专心对待。读者等的只是开工时间，不是烂尾的风险。

对比一下老办法：三本一起接，看起来谁都没等，结果一起写到一半全崩，三份半成品谁也救不回来。那不是高效，是把三份失望同时发货。

这套思路有个名字，叫限流——大白话说，就是宁可让人在门口排队，也不放进来挤塌屋子。银行柜台、医院专家号，都是这个理。**好的系统不是没有等待，而是把等待明码标价，把崩溃挡在门外。**

所以这本书本身也是个例子：它是在「同时只写两本」的规矩下，安安稳稳写完的。排队这件小事，说到底就一句话——**少即是快，稳才能完。**