

矿工的云

GPU 荒里冒出来的新云，靠借来的卡和电，能不能活过折旧

sissi

面向理工背景的好奇者

导读

GPU 荒里冒出来的新云，靠借来的卡和电，能不能活过折旧

2022 年 9 月 15 日凌晨，以太坊完成了一次叫"合并"（The Merge）的升级。一夜之间，成千上万台专门算哈希的显卡失业了——它们原本靠"拼算力换币"吃饭，现在这条路被一行代码堵死。这些矿工手里攥着仓库、电力合同，和一堆还没折旧完的 GPU。

就在同一时刻，另一件事正在烧起来：大模型训练把英伟达的卡抢成了硬通货，老牌云（AWS、Google Cloud 这些）交不出货，交出来的也不好使。荒地里长出一批新公司——[neocloud（新云）](#)，说白了就是"只干一件事的云：把成千上万张 GPU 用超高速网线捆成一台巨型机器，按集群整租给你训模型"。领头的 CoreWeave，前身正是那批失业的以太坊矿工。

这本书想讲清楚一件挺魔幻的事：**一群矿工，靠借来的卡、借来的电，凑出了 AI 时代最抢手的基础设施**。而它们的生意模型，简单到近乎危险——

借钱买卡，再拿卡去抵押借更多钱，买更多卡。

卡是抵押物，长约是现金流，电是命门。这套打法在算力荒里跑得飞快，但它头上悬着一颗定时炸弹：**折旧**。一张 GPU 到底能挣钱几年？账面上按 6 年摊，可要是新一代卡两三年就把它比成废铁，那些借来的钱就还不上。更悬的是，CoreWeave 一大半的收入来自同一个客户（微软，约 62%）——这到底是抱上了大腿，还是把脖子递了过去？

大白话

一句话看懂这本书要回答什么：这些矿工出身的新云，能不能活过它们自己买的卡的折旧期？

所以别把它当成一本"AI 很牛"的行业赞歌。它是一道生存题。全书 14 章，都在绕着一个问打转：**矿工的云，到底能不能活好？**算力荒总会退潮，推理时代的规则和训练时代不一样，老牌云在反扑，最大的客户在自己造芯片——潮水退下去时，谁还穿着裤子？

建议你带着三个问题读：第一，它们创造的是**真价值**（把算力变成一种能买卖、能融资的商品），还是只是踩中了一段短缺的**时间差**？第二，"借卡再抵押"这套金融杠杆，是聪明还是脆

弱？第三，如果最后拼的是电而不是卡，那胜负手到底在谁手里？

读完这 14 章，你应该能自己判断：下次再看到某家 neocloud 融了几十亿、估值翻了几倍的新闻，它到底是在盖楼，还是在借新还旧、跟折旧赛跑。矿工的云照出来的，其实是整个云计算的下一层地基——那层平时没人给你看的地基。

第一章 · 算力荒是怎么烧起来的

先讲一个很具体的场景，这是这两年真实发生在成百上千家公司身上的事。

假设你是一家 AI 创业公司，手里有一个不错的想法：训练一个自己的大模型。你算了算，需要大概一千张最新的 GPU，连续跑上两三个月。你打开亚马逊的云、微软的云、谷歌的云，准备像过去十年租服务器那样——填张信用卡，点几下，机器就到手。

然后你撞墙了。页面上写着“**该机型当前不可用**”。你打电话给销售，对方很客气，告诉你排队吧。排多久？“目前的交付周期大概……十一个月。”你以为听错了。你要的不是明年的算力，是现在就要训的模型，竞争对手可能下个月就发布了。

这不是段子。2023 年，最抢手的那款卡 H100——英伟达当时最强的 AI 训练芯片，一张卡官方定价三四万美元——从下单到拿货，等待时间一度长达十一个月，也就是五十多周。就算你不买、只想按小时租，市场上的租金从早期约每小时 4.7 美元一路被抢到 8 美元以上，还经常无货。整个行业进入了一种奇怪的状态：**钱不是问题，卡才是问题。**

大白话

说人话：过去你缺的是钱，有钱就能买到算力。现在反过来了——你有钱，甚至有很多钱，却买不到、租不到、排不上队。算力从“随时能买的日用品”变成了“要托关系、要排长队的稀缺品”。

为什么会突然缺成这样

要理解这场“算力荒”，得先弄清一件事：**GPU 这东西，需求不是慢慢涨上来的，是在两年里像按了指数键一样炸上来的。**

先解释一下这里的两个词。GPU本来叫“图形处理器”，是显卡里那块芯片，最早是给游戏画面做渲染用的。它的特点是：不擅长一件一件按顺序办复杂的事，但极其擅长同时办成千上万件简单的事——比如同时算屏幕上几百万个像素点的颜色。后来人们发现，训练神经网络恰好也是这种活：本质上就是海量的乘法和加法一起做。于是这块“游戏芯片”被拉来当了 AI 的主力。

训练大模型是什么意思？粗糙地说，就是拿一大堆文本（比如大半个互联网），让一个有几千亿个参数的数学模型反复读、反复猜下一个词、猜错了就微调参数，重复几万亿次，直到它能像模像样地说话。这个"反复调"的过程，就是天文数字级别的乘加运算，正好喂给GPU。参数越多、数据越多，需要的算力就越大——而且不是线性变大，是成倍成倍地翻。

导火索：ChatGPT

2022 年 11 月，OpenAI 放出了 ChatGPT。这件事的意义不在技术本身有多惊天，而在于它第一次让普通人、也让每一个企业老板亲眼看到：原来大模型能干这么多活。于是几乎所有科技公司在同一时间做了同一个决定——**我也要有一个。**

问题在于"同一时间"这四个字。如果需求是错峰的，供应链还能慢慢跟上。但当谷歌、Meta、微软、还有几百家创业公司在同一个季度、抢同一款芯片时，供给方英伟达就算火力全开也来不及。芯片不是软件，不能复制粘贴——它要经过晶圆厂排产、封装、测试，一整条链条上任何一环都不是几周能扩出来的。需求瞬间跳了一个数量级，产能却只能按年爬坡，中间的缺口就是那十一个月的排队。

这个跳跃有多大，看供货方的账本最清楚。英伟达数据中心业务（就是卖这些 AI 芯片的部门）的收入，2023 财年大约 150 亿美元，到 2024 财年跳到 **475 亿美元**，一年涨了两倍多。更夸张的是把这一年拆到季度看：**2024 财年第四季度（也就是 2023 年底那个季度），光这一个季度的数据中心收入就有 184 亿美元，同比涨了 409%**——去年同期还只是它的五分之一。前面那个"一年翻两倍多"的年度数字，正是被这样一个个像坐火箭的季度拱起来的。一家几百亿美元体量的成熟公司，收入能在一年里翻两倍还多，这在商业史上极其罕见——它不是被"用户慢慢变多"推起来的，是被一场集体性的、突然爆发的抢购推起来的。

大白话

打个比方：这就像全城的人在同一个周末决定都要装空调。空调厂再拼命也造不出这么多，于是价格飞涨、排队几个月、黄牛出现。GPU 荒就是这么个逻辑，只不过"空调"是几万美元一张、还得几千张连一起用的顶级芯片。

为什么需求还在往上翻

更麻烦的是，这场抢购不是一波性的，它自己会往上叠。有几股力量在同时推：

- **模型越做越大。**行业里有个被反复验证的经验规律：模型的能力，大致随着参数量和训练数据量的增加而变强。于是大家陷入一场"越大越强、越强越要更大"的军备竞赛，每一代模型需要的算力往往是上一代的好几倍。
- **玩家越来越多。**一开始只有几家巨头玩得起，后来开源模型出现、融资变容易，一大批创业公司、甚至传统企业也都下场自己训、自己调，抢卡的人从几家变成了几百家。
- **光训练还不够，用起来还要更多卡。**模型训好之后，每一次有人跟它对话、让它写代码，都要再消耗一次算力，这叫推理——就是用训练好的模型来实际干活、给出回答。用的人越多，推理吃掉的算力越大。训练是一次性的高峰，推理却是每天都在涨的长流水。

三股力量叠在一起，需求曲线就变成了一条陡峭的、几乎立起来的线。而芯片产能是一条平缓向上的斜线。两条线之间那块越裂越大的空白，就是"算力荒"这三个字的全部含义。

荒里裂开的那道缝

现在回到开头那家 AI 公司。它排不上队，不只是因为芯片总量不够，还有一个更微妙的原因：**老牌云手里就算有卡，也未必愿意、未必能痛快地整块租给它。**

这里先埋个引子，具体的原因留到后面几章展开。你可以先记住一个反直觉的事实：亚马逊、微软、谷歌这些云巨头，是过去二十年为"千千万万个小租户跑网站"这件事精心设计出来的。它们的整套架构——把一台大机器切成很多小份、同一台机器上塞很多客户、按虚拟机计费——在服务无数个小客户时无比高效。但当你要的是**几千张卡紧密连成一台超级计算机、连续几个月独占着跑一个训练任务**时，这套为"切碎"而生的架构反而变得慢、变得贵、变得别扭。

大白话

再打个比方：老牌云像一栋精装修的公寓楼，一间间隔好、水电齐全，适合成千上万个租客各住一间。可你现在要办的是一场需要整栋楼打通、还得在楼里拉满高速传送带的大型工厂生产。你去租公寓，房东要么没有这么大的连片空间，要么给你的报价贵到离谱、交付慢到离谱。

于是市场上同时出现了两个缺口：一边是**芯片本身不够**，另一边是**就算有芯片，把它们高效地组织成一台大机器、按整块租出去的这门生意，老牌云没准备好做。**

凡是有又大又急的需求、而现有玩家填不满的地方，就一定会有新玩家闻着味冲进来。这本书要讲的主角——那批被叫作新云（也叫 GPU 专用云，英文 neocloud）的公司，专门只干“把成千上万张 GPU 组装成超级计算机、整块租给你训模型”这一件事——正是从这道缝里冒出来的。而其中最出名的一家，出身居然是加密货币矿工。

它们凭什么接得住这波？为什么偏偏是矿工？借来的卡和电，又能不能撑过芯片飞快贬值的那几年？这些，是后面章节要一层层拆开的问题。但故事的起点，就是本章这道越裂越大的缺口：**需求在两年里指数式炸开，供给却只能按年慢慢爬——中间那块空白，等着被人填。**

第二章 · 老牌云为什么不够用

先记住一件事：AWS 是为"一百万个小房客"造的

你去 AWS、Azure、GCP 租机器，本质上是在租一间公寓里的一个房间。这三朵老牌云（hyperscaler，指亚马逊 AWS、微软 Azure、谷歌 GCP 这三家规模最大的通用云），从第一天起就不是为"训练大模型"设计的。它们的设计目标是：**让几百万个互不认识的租户，同时在一批物理服务器上跑各自的小活儿，谁也别撞到谁，账单还得算得清清楚楚。**

这批小活儿是什么？是一个卖鞋的网站、一个记账 App 的后端、一段定时跑的报表脚本、一个公司内网的文件服务。它们有三个共同点：单个用得不多、来的时候一阵一阵、而且彼此毫无关系。AWS 这套架构里的每一个聪明设计，都是冲着"伺候好这一百万个小房客"去的。可一旦你把它拿来跑一个要几千张卡拧成一台超级计算机的训练任务，这些聪明设计就一条条变成了绊脚石。这一章就把这几条绊脚石一根根拆开看。

绊脚石一：它把机器"切碎了卖"

老牌云的核心生意，是虚拟机（virtual machine，简称 VM，指用软件把一台物理服务器切成好几台"假装独立"的电脑，每台跑不同租户的活儿）。一台物理服务器有 128 个 CPU 核、几百 GB 内存，云厂商用一层叫**虚拟化管理器（hypervisor，夹在真实硬件和虚拟机之间的一层软件，负责把一台真机切成好几台假机、并分配谁用多少）**的软件，把它切成十几台、几十台小虚拟机，分给不同的人。

大白话

说人话：一栋楼，房东不整栋租，而是隔成一间间小卧室分开租。这样房子不会空着，每间都能收租，房东赚得最多。云厂商干的就是这个——把大铁块切小，卖给尽量多的人。

这层切分不是免费的。每一次虚拟机里的程序想碰一下真实硬件——读网卡、写硬盘、访问显卡——都要经过 hypervisor 这个中间人翻译一道。对跑网站来说，这点开销无所谓，网站本来就慢。但训练大模型时，成千上万张卡每秒要交换海量数据，中间多一道软件翻译，

就是实打实的税。所以后来连 AWS 自己都不得不推出裸金属（bare metal，指不切虚拟机、把整台物理服务器原封不动租给你一个人，中间没有 hypervisor 这层）机型——等于承认："这活儿，我那套切分帮倒忙。"

绊脚石二：它默认"超卖"

老牌云还有一个赚钱法宝叫超卖（overcommit / oversubscription，指卖出去的资源总量比实际拥有的还多，赌大家不会同时全用满）。这跟航空公司超售机票、健身房卖出远超容量的会员卡是同一个道理：因为绝大多数房客大部分时间是闲着的。

你租了 4 个 CPU 核，但你的网站半夜没人访问，这 4 个核就闲着。云厂商精得很，它把这些闲着的核偷偷匀给别的租户用。100 个租户各买 4 核，物理上可能只配了 200 核而不是 400 核——赌的就是没人会同时满载。对跑网站的人来说这没被察觉，因为你要用的时候通常刚好有空闲核可借。

可这套赌博碰上 AI 训练就崩了。**训练任务的特征恰恰是"所有卡、所有核，从头到尾 100% 满载，一跑就是好几周。"**它没有"闲着的时候"可供偷借。更糟的是，它会撞上吵闹的邻居（noisy neighbor，指跟你共用同一台物理机的另一个租户突然大量占用资源，把你的性能拖慢）——你旁边那个租户一使劲，你的任务就卡一下。对网站，卡一下就是页面慢半秒，没人在意；对一个几千张卡的训练，任何一张卡慢一拍，整台超算都得停下来等它。

为什么"等最慢那一个"是致命的

这里要讲清楚训练和跑网站在数学上的根本区别，这是全章最关键的一点。

跑网站是**各干各的**：一百万个用户的请求互不相干，这个慢了不影响那个，整体快慢是"平均值"。而训练大模型是**一起干一件事**：几千张 GPU 算的是同一个模型，每算完一小步，所有卡都要停下来，把各自算出的结果凑到一起对一遍账，再一起走下一步。这个"对账"动作叫全归约（all-reduce，指把成千上万张卡各自算出的梯度数字加总平均、再发回每张卡，让大家保持同一个模型），它每秒要发生成千上万次。

打个比方：一千个人抬一根巨长的钢梁走路，必须齐步。每走一步就要喊一次“预备——起”，所有人同时迈脚。只要有一个人慢半拍，整根钢梁就得停下来等他。一千个人的速度，不取决于最快的人，也不取决于平均速度，而取决于**最慢的那一个**。

工程上这叫“被长尾拖死”：整个训练的节奏，由最慢的那张卡、最慢的那一次数据传输决定。老牌云那套“平均够用就行、偶尔卡一下没关系”的哲学，在这里是灾难——因为你不是要平均快，你是要**没有任何一次慢**。超卖、共享、多租户，每一个都在给“最慢的那一次”添概率。

绊脚石三：它的网络是为“发消息”造的，不是为“抬钢梁”造的

让几千张卡齐步，最吃的是卡与卡之间的网络。老牌云的机房网络，是按互联网的思路建的以太网（Ethernet，我们平时上网用的那套网络技术，靠一层层软件打包、转发数据）：它的设计目标是“让海量互不相干的小消息各自找到路”，容忍偶尔堵车和重发，因为网页慢一下无所谓。

训练要的是另一种东西：InfiniBand（一种专为超级计算机设计的高速网络，让一张卡能几乎不经过 CPU、直接把数据塞进另一张卡的内存里）。差别有多大？在满负载、大家同时抢路的最坏情况下，InfiniBand 能把延迟稳定压在大约 1 微秒（百万分之一秒）级别；而普通以太网一旦堵车，延迟可能从 1 微秒飙到 50 微秒以上——**差的不是一两倍，是几十上百倍**。回到抬钢梁：喊口令的人如果偶尔卡壳几十倍时间，一千个人根本没法齐步。

老牌云不是造不出 InfiniBand，而是它整个机房的物理布局、布线、交换机，都是围绕“多租户、按虚拟机切、灵活调度”这套逻辑铺的。要把成千上万张卡用 InfiniBand 拧成一整台机器，你得从机柜怎么摆、线怎么走、电怎么供，全部重来一遍。这不是打个补丁能解决的，是地基问题。

于是就“又慢、又贵、又别扭”

把上面几条合起来，就明白为什么老牌云在这波算力荒里显得笨重：

- **慢**：虚拟化这层中间人 + 以太网的堵车长尾，让"抬钢梁"总有人掉队，训练效率被拖下来。
- **贵**：老牌云的定价里,打包了一大堆你训练时根本用不上的东西——弹性伸缩、按秒计费、上百种托管服务、全球多机房冗余。这些是伺候小房客的豪华设施,你一个跑三周满载训练的人,为用不上的东西付了钱。
- **别扭**：你想要"给我几千张卡、用 InfiniBand 连成一台超算、整机交给我、别人别碰",而老牌云的整套自助界面、配额、调度器，都是围绕"切一台小虚拟机给你"设计的。想拿到一大坨连在一起的卡，往往要排队、要谈判、还未必给得出连号的整片。

这里有个反直觉的点值得强调：**不是老牌云技术不行，恰恰是它们太成功了**。正因为把"多租户、超卖、按虚拟机切"这套打磨到了极致、赚得盆满钵满，它们的每一寸设计都长成了那个形状。而 AI 训练要的是几乎相反的形状：单租户独占、绝不超卖、整机裸金属、用超算网络焊成一台机器。让一个为"一百万个小房客"优化到骨子里的系统,回头去伺候"一个要独占整栋楼的巨人",不是调个参数的事,是逆着它十几年的肌肉记忆来。

缝隙就在这里裂开了。有人需要"一整台超算",而老牌云只会卖"一间间小卧室"。谁能把裸金属整机、InfiniBand、独占不超卖这套东西直接做成产品端上来，谁就接得住这波需求——那是下一章的主角，一种和老牌云长得完全不一样的新动物。至于它凭什么长成那样、又是谁最先练出了这身本事，后面几章再说。

第三章 · 新云是什么样的动物

先看一样东西：一台机器，还是一堆机器

假设你要训练一个大模型。你手上有一份任务书：一万亿个参数，几十太字节（TB，太字节，1 TB 约等于 25 万本书的文字量）的文本，要在几个月内跑完。你去老牌云上租机器，租到的东西大概是这样的：一台虚拟机，8 张卡，能干活。你想要更大，就再租一台，再一台。租到几百台的时候，问题来了——它们各自为政。

训练大模型这件事，本质上是一群卡在[反复交换草稿](#)。每张卡算一小块，算完要把结果告诉所有其他卡，对齐了再算下一步。一步接一步，一天上百万次。所以这不是"几千台电脑各自跑各自的"，而是"几千张卡必须像一张卡那样思考"。

这就是新云要回答的问题：怎么把成千上万张会发烫的卡，攒成一台机器。它给出的答案有三样，我们一样一样拆。

第一样：按集群卖，不按虚拟机卖

先说老牌云的卖法。[虚拟机](#)（virtual machine，把一台物理服务器用软件切成好几台"假电脑"，租给不同的人）是云计算三十年来的看家本领。一台真机器切成八份，卖给八个跑网站的小客户，谁也看不见谁，这叫[多租户](#)（multi-tenant，多个互不相识的客户共用同一批物理硬件）。这套模式对"几千万个小租户跑网站"是完美的——它把一台大机器的利用率榨到最高，还能[超卖](#)（oversubscribe，因为大家不会同时满负荷，就把资源卖得比实际多一点，像航空公司超售机票）。

但训练大模型的客户不是小租户。他要的不是"切一块给我"，而是"把这一整片都锁给我，别人一个字节都别碰"。

大白话

老牌云的思路是：把一块大蛋糕切成很多小份，卖给很多人。新云的思路是：你要的就是一整个蛋糕房，连烤箱带工人整套包给你，中间不站别人。

所以新云卖的单位不是"一台虚拟机"，而是一个**集群**——比如"512 张 H100，连成一体，包你三年"。这背后有个物理原因：那几千张卡要像一台机器那样协作，就必须知道彼此的确切位置、谁挨着谁、走哪条线最短。这种"编排"（怎么摆放、怎么连线）没法临时拼凑，得整片规划、整片交付。你按集群买，买到的其实是一张**预先设计好的拓扑图**，而不是一堆随机凑数的机器。

第二样：裸金属，把整机直接交到你手里

第二样叫**裸金属**（bare metal，直接把整台物理服务器交给你，中间不隔虚拟化那层软件）。要理解它为什么重要，得先明白虚拟化"贵"在哪。

那层把真机器切成假机器的软件，本身要占用算力和内存来做管理——这叫**虚拟化开销**（virtualization overhead，为了维持"假电脑"的假象而消耗掉的那部分真实性能）。跑网站时这点开销无所谓，反正 CPU 大把闲着。但训练大模型时，卡与卡之间要用一种叫 **GPUDirect RDMA**（让一张卡的显存直接读写另一张卡的显存，绕开 CPU 和操作系统这些中间人）的技术贴着硬件通信，追求的是微秒（百万分之一秒）级的延迟。中间多隔一层软件，就像你想和隔壁桌说话却被要求每句都先写进公文流转一遍——每一步都慢那么一点点，乘以一天上百万步，就是几天甚至几周的差距。

大白话

虚拟机好比酒店客房：方便、隔音、有前台管着，但墙是薄的，隔壁装修你也得忍。裸金属好比把整栋楼的钥匙给你：没有前台抽成，没有邻居，你想把承重墙打通接根光纤都行。训练大模型，要的就是打通墙。

裸金属还解决了一个信任问题。当你要包三年、押上几千万美元的训练预算，你不希望性能被一个看不见的邻居偷走，也不希望云厂商在你眼皮底下偷偷超卖。整机交付意味着**这批卡的性能是确定的、可预测的、全归你**——对一个要精确算成本、算工期的工程团队来说，确定性本身就是产品。

第三样：InfiniBand，把几千张卡缝成一台超算

前两样解决了"卡都归你、没人挡道"。但最难的是第三样：怎么让这几千张卡通信快到像是一张卡。这就是**InfiniBand**（一种专为超算设计的超高速网络，读作"infini-band"，字面意

思"无限带宽")的活。

先给个量级感。普通数据中心机器之间连网，走的是以太网，常见是每秒几十到几百 Gb（吉比特）。而一台标准的 8 卡 H100 服务器，是**每张卡配一条 InfiniBand 专线，每条 400 Gb/s**（这一代叫 NDR），八张卡加起来约 3.2 Tb/s（太比特每秒）冲出机器。这是什么概念——差不多一秒钟能倒完几百部高清电影，而且是每张卡都有这么一条独享的线。

光快还不够，还得"聪明"。前面说过，训练时卡与卡最频繁的动作是 All-Reduce（把所有卡各自算出的梯度加总、再把总和发回每张卡，是分布式训练里最烫手的一步）。传统做法是所有卡先把数据传到一起、加完、再传回去，卡越多，来回越多，越堵。InfiniBand 有个杀手锏叫 SHARP（Scalable Hierarchical Aggregation and Reduction Protocol，让网络交换机自己顺手把数据加总了）——它把"加总"这个动作直接塞进了网络交换机的芯片里。数据在**路过交换机的途中就被加好了**，不用等所有卡都收齐。几十个节点规模上，这能把来回次数从"随卡数增长"压到接近"一次搞定"。

大白话

打个比方。一千个人要凑份子算总账。笨办法：每个人把自己那份报给一个记账员，记账员一笔笔加，队排到天黑。SHARP 的办法：钱在传递的路上，每经过一个中转台就顺手加一次，到终点时总数已经算好了。网络不只是"送信的"，它还"帮你算"。

这三样是配套的，缺一不可。InfiniBand 的低延迟，会被虚拟化那层软件吃掉，所以必须裸金属；裸金属整机加高速网，只有整片规划、整片交付才连得起来，所以必须按集群卖。它们合起来，才把"一堆机器"变成了"一台机器"。

为什么这恰好是训练大模型要的

把三样叠起来看，新云其实是把云计算的默认值反过来做了一遍：

- 老牌云为**无数小租户**优化，新云为一个**巨型任务**优化。
- 老牌云追求**高利用率**（超卖、共享、切片），新云追求**确定性能**（独占、整机、专网）。
- 老牌云卖的是**灵活**（要多少切多少，随时退），新云卖的是一**整台超算**（整片包断，按年锁）。

训练大模型这件事，恰好每一条都站在新云那边：它是一个必须协同的巨型任务，它对性能的确定性极其敏感，它一包就是几个月甚至几年。这不是新云比老牌云"更先进"，而是**它们服务的物种不一样**——一个是给千万个网站当水电，一个是给一小撮超大模型当专用超算车间。

这也解释了新云为什么能在两三年里从零冲到几十亿美元营收。以 CoreWeave 为例，它 2023 年营收约 2.3 亿美元，2024 年跳到约 19 亿美元，2025 年 3 月登陆纳斯达克。代价是收入高度集中——2024 年光微软一家就约占其收入的 62%，微软加 OpenAI 合计约四分之三。这背后的逻辑正是"按集群包断"：少数几个巨头，一签就是天量长约，撑起了大半张桌子。这种集中是福也是雷，后面的章节会专门拆。

但先记住这一章的结论：新云不是"更便宜的 AWS"，它是一种**为了让几千张卡像一张卡那样思考而重新设计的动物**。按集群卖是它的骨架，裸金属是它的皮肤，InfiniBand 是它的神经。三样凑齐，一堆发烫的卡才第一次真正变成了一台超算。

第四章 · 为什么是矿工接住了这波

上一章我们说清了新云是种什么动物：它不按虚拟机切片卖，而是把成千上万张 GPU 用高速网线连成一整台超级计算机整块出租。问题来了——盖这么一台机器，谁有本事？答案有点反直觉：接住这波 AI 算力荒的，很多是**挖矿的人**。这一章要讲清楚，这绝不是巧合，而是因为挖矿这门苦活儿，恰好把干 AI 云需要的三块肌肉都练出来了。

先说清楚：矿工到底在干什么

为了讲明白，得先把“挖矿”这件事拆开。这里说的不是挖煤，而是 GPU 挖矿——用大量显卡去算一种数学题，谁先算出来谁就拿到加密货币奖励。在 2022 年之前，最赚钱的是挖以太坊（Ethereum，一种加密货币，当时用“拼算力”的方式发行新币）。

大白话

你可以把它想成：一屋子显卡二十四小时不停地掷骰子，谁先掷出一个符合条件的数，谁就能领一笔钱。掷得越快（算力越大）领钱概率越高。于是矿工的全部生意逻辑就变成一句话：**怎么让尽可能多的显卡，尽可能便宜地、尽可能不停地转下去。**

注意这句话里藏着三件苦差事。第一，显卡耗电惊人，电费直接吃掉利润，所以你必须搞到**全世界最便宜的电**。第二，你得有地方摆这些卡、给它们通风散热供电，也就是得**快速盖出机房**。第三，几千上万张卡挤在一起日夜满载，坏卡、过热、掉线是常态，你得有本事**大规模伺候这一屋子发烫的卡**。

现在把这三件事和上一章的 AI 云需求对一下——你会发现它们几乎是同一张清单。这就是本章的核心：**挖矿练出的三块肌肉，正好是 AI 云的入场券**。下面一块一块看。

肌肉一：搞到便宜电

先建立一个直觉：**大规模算力的头号成本不是芯片，是电**。一张顶级 AI 显卡（比如英伟达的 H100）满载时约耗 700 瓦，一整台几千卡的集群加上散热，动辄要吃掉几十兆瓦——

兆瓦 (MW) 是功率单位，1 兆瓦大约是一千户普通家庭同时开空调的用电量。几十兆瓦，就是一个小镇的电。

矿工在挖矿时代，利润薄得像刀片，电价每度差一分钱，一年下来就是生死线。所以他们被逼成了"找电专家"：哪里有水电站的富余电、哪里有电网用不完的谷电、哪里有别人不要的能源，他们就把机器搬到哪里。

最极端的例子是 Crusoe（一家从挖矿转做 AI 数据中心的公司）。油田打油时会伴生大量天然气，运不出去，只能就地点火烧掉——行话叫放空燃烧 (flaring)，白白烧掉还污染。Crusoe 干的事是：把这些本来要烧掉的气接进发电机，就地发电，就地挖矿。等于捡了一堆免费的、别人当垃圾的能源。它靠这套在美国好几个州和阿根廷铺了四百多个模块化机房、两百多兆瓦。

大白话

为什么这条肌肉对 AI 云值钱？因为 AI 训练同样是"电老虎"，而 2024、2025 年真正卡脖子的已经不是买不到卡，而是**拿不到电、并不上网**（这本书第十章会专门讲）。一个练过"全球找便宜电"的团队，和一个从没有为电发过愁的软件公司，谁更可能在两年里搞定几百兆瓦？答案很清楚。

肌肉二：快速盖机房

光有电还不够，卡得有地方住。这里要澄清一个外行的误会：**数据中心不是盖个房子摆上机架就行**。它是一套精密的物理工程——要把几十兆瓦的电稳稳送进来、要在卡满载发热时把热量抽走、要有备用电源保证不断电、要有网络布线。这些活儿又慢又重，跟写软件是两个世界。

而矿工天生就是干这个的。挖矿利润随币价剧烈波动，谁能**更快把新机房建起来、更快把卡通上电开始转**，谁就能在行情好的时候多赚。所以矿工被市场逼出了一种"抢工期"的建设能力：不追求豪华，追求快和便宜——模块化、标准化、哪里有电有地就地起一片。

典型的路径是 CoreWeave（如今最大的新云之一，2025 年 3 月在纳斯达克上市）。它的前身叫 Atlantic Crypto，2018 年前后是北美最大的以太坊矿工之一，在七个数据中心里塞了数千张英伟达显卡，还自己捣鼓散热方案。**建机房、供电、散热、塞卡——这套本事它**

在挖矿时代已经练了好几年。2019 年它嗅到风向不对（以太坊迟早要抛弃 GPU 挖矿），干脆掉头去给缺卡的机器学习研究者租显卡。到 2025 年，它已经运营着 40 多个数据中心、约 850 兆瓦在用电量、约 25 万张 GPU。

这里的因果链很干净：**先有了盖机房、供电、散热的硬功夫，转型时才只需要换"客户"，不需要换"本事"。**一个纯软件出身的团队要从零学会这些重工程，起步就慢了好几年——而在算力荒里，几年就是错过整个窗口。

肌肉三：大规模运维一屋子发烫的卡

第三块肌肉最不起眼，却最难。想象一个房间里塞了上万张显卡，全部满载、日夜不停、烫得能煎蛋。在这种状态下，**坏卡不是意外，是日常**——每天都有卡过热降频、掉线、彻底罢工。

对挖矿来说，一张卡停转就是一份钱在流失，所以矿工被迫练出一整套"大规模运维"的手艺：怎么快速定位哪张卡出了问题、怎么在不停整机的情况下换掉坏卡、怎么调风道和功耗让一屋子卡长期稳定地烫着转。这是一种蓝领式的、脏活累活的运营能力，写代码写得再漂亮也替代不了它。

大白话

为什么这条对 AI 尤其致命？训练一个大模型，是让几千张卡**作为一个整体连续跑几周甚至几个月**，中间只要有一张卡挂了、整台"超算"就可能卡住，几周进度付诸东流。所以 AI 集群对"卡不能随便掉、掉了要秒级换上"的要求，比挖矿还苛刻。一个练过大规模伺候发烫显卡的团队，正好接得住。

把三块肌肉拼起来看

现在退一步，把这三样并排放：搞到便宜电、快速盖机房、大规模运维发烫的卡。你会发现一个惊人的事实——**这既是挖矿的全部苦活，又是 AI 云的入场券，两张清单几乎完全重合。**

这就解释了为什么是矿工接住了这波，而不是别人。它不是运气，是能力错配的结果：

- **老牌云**（AWS 那种）有的是软件、调度和多租户经验，但它们从没被逼着去油田捡免费天然气、去抢工期盖重资产机房；这些恰恰是新云最需要的。
- **纯 AI 创业公司**懂模型、懂软件，但让它去搞几百兆瓦电力和上万张卡的物理运维，等于从零建一家重工业公司。
- **矿工**手里的正好是那套“又脏又重、别人不愿干也干不快”的物理功夫，唯一要换的是把机器上跑的东西从“掷骰子挖币”换成“训练模型”。

换句话说，AI 算力这门生意，表面看是芯片和软件的故事，骨子里是一门**能源加房地产加重运维**的实体生意。谁把这套实体功夫练熟了，谁就有资格入场——而在 2019 到 2022 那几年，恰好是矿工把这套功夫练到了肌肉记忆的程度。

当然，肌肉对了，只是有了入场券，不等于稳赢。矿工手里有能力，但真正把这些能力猛地推向 AI 市场，还需要一个触发事件——2022 年 9 月，以太坊一夜之间抛弃了 GPU 挖矿，让这三块肌肉突然无处可用。那一天发生了什么、又是怎么把整个行业的显卡和人一起推向 AI，是下一章的事。

（本章数字为公开报道的量级：CoreWeave 前身 Atlantic Crypto 曾是北美最大以太坊矿工之一，2019 年转型做 GPU 云，2025 年 3 月纳斯达克上市；其规模约 40 多个数据中心、约 850 兆瓦、约 25 万张 GPU。Crusoe 以油田放空天然气发电起家，铺设两百多兆瓦模块化机房后转向 AI。具体数字随时间变动，此处取合理量级。）

第五章 · 以太坊合并那天，矿工失业了

一夜之间，几百万张卡没活干了

2022 年 9 月 15 日，凌晨。以太坊这条区块链完成了一次早就预告好的手术，工程师们管它叫 The Merge（合并）——把原来靠显卡算数字挖矿的机制，换成一套靠"押钱"来记账的新机制。手术很成功，成功到什么程度？以太坊全网的耗电量，一夜之间掉了大约 **99.95%**。而在电表的另一端，全世界几百万张正在轰鸣的显卡，突然失去了它们唯一的工作。

大白话

翻译成成人话：以前想在以太坊上记一笔账、拿一点奖励，全世界的矿工得比赛谁的显卡算得快，谁先算出一道没意义的数学题谁记账。这叫"拼算力"。合并之后，规则改成"拼谁押的钱多"——你锁 32 个以太坊进去当保证金，系统就随机让你记账。记账不再需要答题了，于是那道"没意义的数学题"消失了，专门用来解题的显卡，也就一起失业了。

要理解这一天为什么重要，得先搞清楚被淘汰掉的到底是哪套东西。

先分清两种"挖矿"

外行常把"挖矿"当成一件事，其实市面上一直有两种挖法，用的硬件天差地别。

- **比特币挖矿**：用的是 ASIC——一种"专用集成电路"，芯片被焊死了只会算比特币那一种题，别的什么都干不了。合并跟它没关系，比特币至今还在用显卡……不，用 ASIC 挖。
- **以太坊挖矿**：用的是 GPU——就是游戏显卡、AI 训练用的那种"图形处理器"，一块能干很多种并行计算的活。以太坊当年故意把算法设计成"抗 ASIC"，逼大家用通用显卡挖，好让普通人也能参与。

这个设计上的小心思，埋下了一个巨大的伏笔：**以太坊矿工手里攒的，恰好是一堆通用 GPU，而不是只会挖矿的废铁**。合并那天，比特币矿工的 ASIC 除了挖矿一无是处，只能当

废品；但以太坊矿工的显卡，本质上和后来训练大模型用的卡是同一族生物——只是型号老一点。

失业的不只是卡，是一整套"重资产肌肉"

如果只是显卡闲置，那不算大新闻，二级市场消化一下就完了。合并之后确实发生了这件事：矿工恐慌抛售，二手显卡价格腰斩，英伟达当年顶配的 RTX 3090 Ti 发布时标价 1999 美元，半年后就有人挂到 1100 美元以下甩卖，商店柜台里堆满没人要的卡。

但真正值钱、也真正被这一天推向 AI 的，不是卡本身，而是矿工为了让这些卡昼夜运转，被迫练出来的一整套本事。前一章讲过这三块肌肉，这里再收拢一下，因为合并那天，它们全都突然空转了：

- **拿到便宜大电的能力。**挖矿的利润薄得像纸，电费是生死线，所以矿工全是"电力猎人"——专挑水电富余的山区、油田烧掉的伴生气、电网用不完的低谷电，签几十上百兆瓦的长约。兆瓦 (MW) 是功率单位，一兆瓦约等于一千户普通人家同时开空调的用电量；而一个像样的 AI 数据中心张口就要几十上百兆瓦。这种"能签到大电"的本事，普通创业公司要磨好几年。
- **快速盖机房的能力。**矿场不是写字楼，是一排排通风散热的钢棚，能在几个月里从一块空地变成塞满机器的厂房。AI 数据中心要的也正是这个：快、密、能扛热。
- **大规模伺候发烫硬件的能力。**几万张卡满负荷跑，坏卡、过热、供电波动是家常便饭。矿工天天跟这个死磕，练出了一套"让一大片发烫的芯片持续不宕机"的运维手艺。

大白话

换句话说：合并那天真正闲下来的，是"便宜的电 + 现成的机房 + 会伺候热硬件的团队"这一整套东西。卡只是其中最不值钱的一环。

时机好得像剧本：一边刚失业，一边正缺人

历史有意思的地方在于时点。以太坊合并是 2022 年 9 月；而就在两个月后，2022 年 11 月，ChatGPT 上线，全世界对 GPU 的需求像被点着的干柴一样烧起来（这正是第一章讲的那把火）。

于是出现了一个近乎荒诞的错位：一边，一批人手握大电、机房、运维团队和成堆 GPU，却突然没活干；另一边，一批 AI 公司抱着钱排队租卡，却租不到、排不上。供给和需求各自站在悬崖两侧，中间缺一座桥。矿工，恰好就是那座桥——他们要做的，不过是把手里的东西从"算没意义的数学题"改成"算大模型的矩阵乘法"。对显卡来说，这两件事本质上是同一种运算：大量数字的并行乘加。

需要说清一个容易被神话的细节：并不是矿工们在合并当晚才灵机一动。真正提前上岸的那家，早在三年前就看懂了。

CoreWeave：提前三年跳船的那一个

这本书后面会反复提到 CoreWeave——一家 2025 年 3 月刚在纳斯达克上市、如今被当成"新云"代名词的公司。它的出身，正是以太坊矿工。

它 2017 年成立，最早的名字叫 **Atlantic Crypto**，三个搞大宗商品交易的人把挖矿的爱好做成了生意，在新泽西的机房里用 GPU 挖以太坊。关键的一步发生在 **2019 年**：那时以太坊转 PoS 的路线图已经摆在明处，聪明人都看得出 GPU 挖矿早晚归零。与其等死，他们干脆改名 CoreWeave，把同一批显卡、同一批机房、同一套运维，转去对外出租算力——最初的客户是做影视特效和渲染的。

大白话

划重点：他们赌对的不是"AI 会火"，而是"我这套硬件迟早要找个挖矿之外的活干"。AI 的爆发是后来白送的运气，但硬件和团队早就备好了。等于提前三年把船划离了正在下沉的码头，结果新码头恰好停在了金矿边上。

数字能说明这运气有多离谱：CoreWeave 2023 年营收约 **2.3 亿美元**，2024 年直接跳到约 **19 亿美元**——一年翻了大约八倍。一家挖矿出身、几乎没人听说过的公司，就这么"从天而降"，冲进了 AI 基础设施的第一梯队。

所以合并那天没有立刻造出一个 CoreWeave；CoreWeave 是提前跳船、又赶上大潮的极少数。合并真正干的事，是把**剩下那一大批还在船上的矿工**，连人带卡带电一起，硬生生倒进了刚刚爆发的 AI 算力市场——有的转型成功，有的抛售离场，有的被后来的新云收编。

但先别急着鼓掌

把矿工写成"天选之子"很爽，可这一章不能停在这里。有几个刺，从合并那天起就扎进了这门新生意，后面每一章都会去挑：

- 矿工手里那批合并前的卡，大多是挖矿世代的老型号，跑当时最新的大模型训练已经吃力。真正能卖钱的 AI 算力，靠的是英伟达 A100、H100 这些新卡——**而这些新卡得花大钱重新买**。于是就有了第六章的核心动作：借钱买卡，再拿卡去抵押借更多钱。
- 显卡这东西一两年就被下一代碾压（A100→H100→H200→B200 一代接一代），跌价飞快；可借来的债是硬的、必须还的。这颗折旧炸弹，留到第八章拆。
- 会挖矿不等于会做云。把几万张卡用高速网络连成"一台超级计算机"稳定地租出去，是另一门手艺——第三章讲过它的样子，后面还会看到不是每个矿工都学得会。

合并那天，最准确的说法不是"矿工赢了"，而是：**命运替一群搞重资产的人，强行按下了转行键**。它们被推上了一张牌桌，桌上摆着 AI 时代最抢手的筹码；但入场费要靠借债来付，而筹码本身正在以肉眼可见的速度贬值。这场赌局怎么下、谁能活着走出来，是这本书接下来要一层层揭开的东西。

第六章 · 借钱买卡，再拿卡去借钱

上一章我们看到，以太坊合并那天，一大批矿工手里的显卡一夜之间从“印钞机”变成“取暖器”。CoreWeave 提前三年跳车，把矿卡换成了给 AI 训模型的数据中心。但这里有个绕不过去的坎：**它们其实买不起卡**。这一章我们就把这层窗户纸捅破——看清 CoreWeave 这类公司真正的商业模式，不是“卖算力”，而是一台把显卡变成抵押物、再拿抵押物去借更多钱的金融机器。它既是加速器，也是火药桶。

先算一笔让人头皮发麻的账

一张英伟达 H100 显卡，2023 年市场价大约三万到四万美元一张。你要组一台能训大模型的超算，不是买几张，是买上万张。一万张 H100 就是三四亿美元——这还只是卡钱，不算把它们连起来的高速网络、机房、电、冷却。

问题来了：CoreWeave 2023 年全年营收才约 2.3 亿美元。**一台超算的采购价，就超过了它一整年的收入**。它显然没有几十亿现金躺在账上等着买卡。那钱从哪来？

大白话

就像你月薪一万，却要买一套五百万的房子。你不可能攒够钱再买，你得先借。CoreWeave 面对的是同一道题，只不过数字后面多了好几个零，而且它买的不是会升值的房子，是会飞快贬值的显卡。

那笔关键的钱：拿显卡当抵押物借钱

2023 年 8 月，CoreWeave 干了一件当时听起来很新鲜的事：它借到了 **23 亿美元** 的债，领投的是两家做另类投资的机构——[Magnetar Capital](#)（一家对冲基金，擅长结构化的、复杂的债权投资）和 [Blackstone](#)（黑石，全球最大的另类资产管理公司之一，管着上万亿美元）。跟投的还有 Coatue、DigitalBridge、贝莱德、太平洋投资（PIMCO）、凯雷这一长串名字。

关键不在借了多少，而在**拿什么做抵押**。这笔钱的抵押物，就是那些英伟达 H100 显卡本身。

大白话

抵押（collateral）就是：你借钱，得押点值钱的东西给债主。你还不上，债主就把东西拿走卖了抵账。房贷押房子，车贷押车。CoreWeave 这笔债，押的是显卡。

这就是那句被反复提起的“金融创新”：**第一次有人把英伟达 GPU 当成像房子、像卡车一样的可抵押资产**。听起来天经地义，其实很反直觉——过去没人愿意接受一堆芯片当抵押品，因为芯片会过时、会贬值、砸手里很难变现。是 AI 算力荒改变了这个判断：在 2023 到 2024 年，一张 H100 租出去的年租金，可能接近它买入价的一半甚至更多。债主一算，就算 CoreWeave 垮了，把这些卡收回来转租或转卖，也能收回不少本金。于是他们敢借。

这台机器的核心齿轮：DDTL

这笔债用的工具叫 DDTL（delayed draw term loan，延迟提取定期贷款）。名字拗口，机制却很聪明。普通贷款是“啪”一下把钱全给你，你从第一天就得为全额付利息。DDTL 不是——它是债主先给你批一个额度，你要用多少、什么时候用，自己控制。

大白话

把 DDTL 想成一张有额度上限的信用卡，专门用来买卡。你签下一个客户订单，需要买五千张显卡了，才从这张卡里刷出对应的钱去买；没订单就不刷，不产生利息。买来的显卡当场就变成这笔刷卡额度的抵押物。

这个设计精准咬合了 CoreWeave 的生意节奏。它的做法是：**先拿到一个客户的长期算力合同，再去借钱买卡来交付这个合同**。比如它和微软签了几年的大单，就拿这份合同去撬动一笔 DDTL——债主看到“有微软兜底的稳定现金流”，才愿意把钱借出来。所以每一笔债背后，往往捆着一份具体的客户合同。卡是抵押物，合同是还款来源，两样一起押。

为什么说是“借钱买卡，再拿卡去借钱”

现在把飞轮转起来，你就看懂了整个模式：

1. 签下一个客户的长期算力合同（比如微软）；
2. 拿这份合同 + 将要买的显卡，做抵押，借一笔 DDTL；
3. 用借来的钱买英伟达显卡，组成超算，交付给客户；
4. 客户按月付租金，这现金流用来还债、付利息；
5. 规模做大了、手里的卡和合同更多了，再拿它们做抵押，借下一笔更大的债，买更多卡。

你看这个循环——**卡既是花钱买来的成本，又是能借钱的抵押物**。买得越多，能抵押的资产越多，能借的钱越多，又能买更多。2024 年 5 月，同样是 Blackstone 领投、Magnetar 联合，CoreWeave 又拿下一笔 **75 亿美元** 的债务融资——比第一笔翻了三倍多。到 IPO 前后，它背的债从 2024 年的不到 80 亿美元，一路滚到后来的两百多亿美元量级。

大白话

说白了，CoreWeave 表面上是家云计算公司，骨子里更像一家**专门经营显卡的融资租赁公司**：用别人的钱买卡，把卡租出去赚差价，赚来的钱先还债。它赚的是“租金 - 利息 - 折旧”这个夹缝里的钱。

加速器：为什么这套打法能起飞

这个模式的威力在于，它把老牌云那种“用自有利润慢慢扩张”的天花板拆掉了。AWS 建数据中心靠的是自己赚的钱，扩张速度受利润约束。CoreWeave 不受这个约束——只要有客户合同、有人肯借钱，它就能几乎无上限地堆卡。算力荒里谁能最快把卡摆出来交付，谁就赢，而杠杆恰好让它跑得最快。它两年里营收从 2.3 亿冲到约 19 亿美元，靠的就是这台借钱机器。

更妙的是，随着 AI 越来越热，债主对 GPU 抵押品的信心也在涨。到 2026 年，CoreWeave 甚至拿到了**投资级评级**的 GPU 抵押贷款（穆迪给 A3）——这在几年前不可想象。这意味着“显卡是靠谱抵押物”这个观念，已经被主流金融机构盖章认可了。GPU 从一堆易过时的电子零件，被硬生生做成了一类金融资产。

火药桶：同一根杠杆，反过来也能炸

但杠杆是把双刃剑。让它飞起来的三个东西，也正是随时能引爆的三根引信。

引信一：折旧比债务软，债务比折旧硬。显卡会飞快过时——A100 到 H100 到 H200 到 B200，基本一两年一代，新卡一出，旧卡的租金立刻往下掉。这就是下一章的主角。CoreWeave 在财报里按大约 **6 年**来摊销显卡的价值（折旧）。做空机构 Kerrisdale 在 2025 年 9 月的报告里死咬这一点：如果显卡实际只能用 4 到 5 年就沦为鸡肋，那折旧一提高，利润就从大约 20% 的经营利润率被打到接近零甚至为负。

大白话

抵押物在贬值，欠的债却一分不少。你押着一堆卡借了钱，两年后新卡出来，你手里的卡只值原来的一半——可你欠债主的本金还是原来那么多。抵押物缩水、债务不缩水，这个缺口就是火药。

引信二：利息不便宜，而且是浮动的。早期这些 DDTL 的实际利率高达 11% 到 15%——第一笔 23 亿美元的债，浮动利率一度接近 15%，2028 年还有一大笔"气球式"到期本金要一次性还上。气球贷就是平时只还小头、最后砸下来一个大额本金。只要客户租金的现金流稍有断档，还债的节奏就会立刻紧张。

引信三：全押在英伟达和少数几个客户身上。抵押物是英伟达的卡，投资人里也有英伟达（它同时投了 CoreWeave、Lambda、Nebius 等好几家新云），供应商、投资人、甚至间接的需求方高度重叠。批评者刻薄地把这叫"互相买货抬价"的闭环。而还款靠的客户又极其集中——微软一家就占了 CoreWeave 2024 年收入的约 **62%**，微软加 OpenAI 合计约四分之三。抵押物、资金、客户全都押在很窄的几个点上，任何一个点晃动，整台机器都会跟着晃。这个"一个客户撑起半壁江山"的风险，第九章会专门拆。

这一章的一句话

CoreWeave 真正的发明，不是更好的云，而是**把显卡变成了可以反复抵押、循环加杠杆的金融资产**——借债买卡，再拿卡去借更多钱。这台机器在算力荒和涨租金里转得飞快，让一家营收几亿的公司撬动了几百亿的资产。但它转起来的前提，是显卡保值、利率可控、客户不跑。一旦显卡贬值的速度超过债务偿还的速度，这台加速器就会变回火药桶。而显卡贬值这件事——恰恰是物理规律里最确定会发生的一件事。下一章，我们就去拆那颗名叫"折旧"的定时炸弹。

第七章 · 它们到底创造了什么新价值

前面几章我们看清了新云是怎么长出来的：从矿工的机房里冒头，借债买卡，再拿卡去抵押借更多钱。但这些还只是"手段"。一个爱因果的读者会追问：**说到底，它们到底做出了什么老牌云做不出、也不愿意做的新东西？**如果答案只是"GPU 租得比 AWS 便宜一点"，那这本书就不值得写——便宜这件事，价格战一打就没了，谈不上什么护城河，也算不上给产业创造了新价值。

这一章我想说清楚：新云真正的创造，不在硬件，而在**它把"算力"这种东西，重新组织成了一种可以签长约、可以抵押、可以按整台超算包断、还能贴着芯片一生一世去伺候的资产。**这四件事，老牌云一件都没认真做。我们一件件拆。

先说清楚：老牌云卖的到底是什么

要看出新东西新在哪，得先看旧东西旧在哪。老牌云（AWS、Azure、GCP 这类通用公有云）的生意，本质是把一大堆服务器切成小块，按小时租给几百万个互相不认识的租户。你半夜跑个脚本，用两小时，用完就还；它把同一台物理机上的空闲，转手租给下一个人。这套模式的灵魂词是弹性——想用就有，不用就退，按用量结账。

大白话

说人话：老牌云像共享充电宝。它赌的是"没人会一直霸占着不放"，靠成千上万人的用完就还，把一块电池的价值榨到最高。它最怕的，就是有个大客户说"这一万块电池我全包了，接下来五年谁都别碰"。因为那样一来，弹性没了，超卖的空间没了，它引以为傲的那套调度魔法就全废了。

而 AI 训练，恰恰就是那个"我全包了"的大客户。训一个大模型，要成千上万张卡连成一体、连续几个月不停机地算。这不是"弹性"的活，是"包场"的活。老牌云的整套定价、调度、财务模型，都不是为包场设计的。新云做的第一件事，就是把生意从"按小时零卖"改成"按整块包断"——这背后不只是卖法变了，是它把算力变成了一种可以签长约的东西。

新价值一：把算力做成能签"类期货长约"的资产

这是最反直觉、也最重要的一点。新云卖 GPU，主力合同不是"用多少付多少"，而是一种叫 take-or-pay 的条款——直译"要么用、要么照付"。意思是：客户提前锁定一批 GPU、一段年限，**无论实际用没用满，钱都得照合同付**。

大白话

这跟买期货、跟包机、跟长期租楼是一个逻辑。航空公司怕明年油价暴涨，今天就锁个价把明年的油买下来；不管到时候油便宜还是贵，都按今天签的价格结。AI 公司怕的不是油价，是"明年根本抢不到卡"。所以它宁可今天就签一张五年的合同，把算力锁死——用不满也认了，总比到时候无卡可用、模型停在半路强。

CoreWeave 这类公司几乎全部收入来自这种多年锁定合同。以它和 OpenAI 2025 年 3 月那笔为例，公开口径是约 **119 亿美元、五年**的算力承诺，带 take-or-pay。到 2026 年，它对外披露的合同储备（backlog，已签字、未来才逐年确认成收入的订单总额）滚到了千亿美元量级——而它 2024 年的实际收入不过约 19 亿。也就是说，账上躺着的是未来好几年的确定性订单，收入还没发生。

这才是真正的新东西：**算力从一种"当下的服务"，被改造成了一种"未来的合约"**。它有交付年限、有违约条款、有可预测的现金流曲线——长得越来越像一张金融票据，而不像一次云主机的开机。老牌云也有折扣预留实例，但那是省钱工具，本质还是弹性；新云的长约是反过来的，它把"不弹性"本身当成了商品：客户花钱买的，正是那份"这批卡未来几年一定是我的"的确定性。

新价值二：先卖后建，把风险扛在合同后面

顺着长约往下想，一个工程师会立刻问：借那么多债买卡，万一卖不掉呢？新云的答案是——**不签合同就不建**。它的典型打法是"先拿到 take-or-pay 长约，再去买卡、盖机房"。合同在前，重资产在后。

这一步之所以关键，是因为它把整条链子接通了。回想第六章：这些公司拿 GPU 去抵押借钱。可银行凭什么敢用一堆两年就贬值的卡做抵押、放出几十亿贷款？答案就在这里——**抵押物是卡，但真正让银行安心的，是那张 take-or-pay 长约背后确定的现金流**。卡会贬

值，但"微软/OpenAI 未来五年必须照付"的合同不会。于是长约先给硬件兜了底，硬件再拿去抵押，钱借出来买更多卡，去接下一张长约。

大白话

说人话：这就像一个包工头，从不先囤钢筋水泥。他先拿到甲方"这栋楼我一定要、五年租金先签死"的合同，拿着这张合同去银行贷款，贷来的钱才去买料盖楼。银行看的不是那堆钢筋能卖几个钱，而是甲方那份跑不掉的租约。新云把 GPU 玩成了这套地产逻辑。

新价值三：按"整台超算"包断，而不是按虚拟机切

光有长约还不够，客户买到手的东西本身也不一样。老牌云给你的是一台台虚拟机，要自己想办法把它们连起来——而多租户环境里，网络是共享的、会互相打架。新云卖的是裸金属集群：整台物理机直接交给你，中间不隔一层虚拟化损耗；再用 InfiniBand（一种超低延迟的专用高速网络，可以把成千上万张卡连成一台逻辑上的巨型计算机）把这些卡编织在一起。

为什么这对客户是真价值？因为训大模型时，几千张卡必须像一个人的左右手一样同步——这一步算完，所有卡要几乎同时把结果对齐，再进下一步。只要网络里有一处堵，全体等它，整台超算的效率就塌了。老牌云的多租户共享网络，天生做不到这种"整体同步"；新云把整块集群、整套网络**不切分、不超卖地整体交付**，让上万张卡真正当成一台机器用。这不是便宜不便宜的问题，是"能不能训得动"的问题。

新价值四：贴着 GPU 的一生做运营

最后一件，容易被忽略，却是新云和老牌云气质上最不一样的地方。GPU 迭代极快——A100、H100、H200、Blackwell，大约一两年就换一代，新卡一出，旧卡的市价立刻被压下去。这对财务是灾难（下一章专讲这颗折旧炸弹），但对**运营能力**反而是护城河。

新云做的是"贴着芯片一生一世去伺候"：新卡刚出，抢在别人前面拿到货、当天上架卖最贵；卡正当年，靠密集运维把它 7×24 塞满、发烫也不宕机；卡老了，就调去跑对速度不那么敏感的活，或者转租、二手处置，把最后一点残值榨出来。这套"从上架到退役全周期压榨"的手艺，正是矿工的老本行——他们伺候发烫的卡、跟电费死磕、把机器利用率抠到极限，本就干了很多年。

说人话：老牌云像开五星级酒店，房间装修好、标准统一，卖的是稳定和服务。新云更像开赛车队：拿到新引擎第一时间榨出最高转速，引擎过气了就降级去跑普通赛道，一直压到报废前的最后一圈。它赚的是"把每一张卡在它短暂的黄金期里用到极致"的运营差。

这里也埋着全书最大的争议。CoreWeave 财报里假设 GPU 能用约六年来摊折旧，可它签的长约平均只有约四年——**卡还没折完，租约先到期了**。做空者（如 Kerrisdale）咬的就是这一点：万一四年后卡还没赚回本、客户又不续约，谁来填这个洞？这是它"贴生命周期运营"这门手艺的成色考题，我们下一章展开。

把这四件事收拢成一句话

回到开头那个问题：新云到底创造了什么新价值？不是更便宜的 GPU 出租。是它把一堆快速贬值的芯片，重新组织成了**四样老牌云没做的东西**：

- **可签长约的算力**——用 take-or-pay 把"未来几年的确定算力"做成一张类期货合约，卖的是确定性。
- **可抵押的资产**——先签约后建设，用长约现金流给硬件兜底，让 GPU 能进金融机器去撬动更多资本。
- **可整体包断的超算**——裸金属加 InfiniBand，把上万张卡不切分地当一台机器交付，解决"训不训得动"。
- **贴生命周期的运营**——从新卡上架到旧卡退役全程压榨残值，把矿工的运维肌肉变成利润。

对客户，它给的是"未来一定有卡、且卡真能连成超算"的确定性；对产业，它做的是一件更深的事——**让算力第一次长得像电力、像地产那样，可以被签约、被抵押、被金融化和实体化地经营**。这也正是这本书最后一章要回到的主题。但确定性是有代价的：所有这些新价值，都架在"卡不会太快废、客户不会跑、电和债都续得上"这几根柱子上。其中最硬、最要命的一根，就是折旧——下一章，我们去拆那颗定时炸弹。

第八章·折旧这颗定时炸弹

一颗定时炸弹，藏在会计表的最不起眼处

前面几章我们看到了一台漂亮的机器：借钱买卡，把卡抵押出去再借更多钱，把 GPU 变成一种能生息、能包断、能上市的金融资产。这台机器跑起来的时候闪闪发光。但所有靠杠杆运转的机器，都有一个共同的软肋——**你借的钱是硬的，你买的东西是软的。**

这一章我们只掰扯一件事：折旧，就是一件东西随着用旧、随着被新款碾压，价值一点点掉下去的过程。会计上，你花大钱买的设备不能在买下来那一年一次性算成成本，而是要摊到它“能用几年”里去，每年扣掉一块。问题来了——GPU 到底能用几年？这个看似枯燥的假设，恰恰是整台机器会不会自己塌掉的引信。

大白话

说人话：你贷款买了一辆车跑网约车，银行的月供是死的，一分不能少。但如果这辆车三年后就没人愿意坐、跑不出钱了，而你的贷款要还六年——后面三年你就是在给一堆废铁还债。GPU 这门生意，怕的就是这个。

先算一笔账：一张卡的赚钱窗口有多长

英伟达的换代节奏，是理解这一切的钥匙。2020 年 A100（Ampere 架构，上一个训练主力卡），2022 年 H100（Hopper 架构，ChatGPT 时代的当红卡），2024 年 B200（Blackwell 架构，新一代旗舰）。差不多两年一大代。而英伟达 CEO 黄仁勋已经公开说，公司要转成“一年一个节奏”。换句话说，你手里这张最贵的卡，**十八到二十四个月后就会被一张更快、更省电、单位算力更便宜的新卡盖过去。**

这不像挖掘机。一台挖掘机十年前和今天挖土的效率差不多，老了照样能干活。但 GPU 是算力，算力是拿来比的：当新卡训练同一个模型只要一半时间、一半电费，客户为什么还要按原价租你的老卡？老卡不是坏了，是**相对变贵了**。它还能通电，只是没人愿意为它付原来的价钱。

所以真正的问题不是"这张卡能亮几年"，而是"这张卡能**按高价**租出去几年"。这个"赚钱窗口"，业内估计大约就是三年上下——最贵、最抢手的头两年赚回大头，第三年开始降价甩尾。

六年，还是三年？争议就卡在这个数上

现在把会计假设摆进来。CoreWeave（本书主角，矿工出身、后来做 GPU 云、2025 年在纳斯达克上市的公司）在财报里，把这些技术设备（GPU、网络、存储都算在内）按**约六年**的年限做直线折旧——2023 年初，它还专门把这个年限从五年调到了六年。

大白话

直线折旧，就是把一张卡的成本平摊到七十二个月，每个月扣掉七十二分之一当成本。年限拉得越长，每个月扣的越少，账面上的利润就显得越好看。把五年改成六年，一年少扣的折旧就是好几亿美元——利润凭空多出来一大块，但卡还是那批卡。

做空机构 Kerrisdale（一家专门找上市公司毛病、赌它股价会跌的投资公司）盯的就是这里。它的逻辑很直白：硬件的真实寿命就三到五年，何况现在换代更快，凭什么按六年摊？如果把年限从六年缩到四年，每年的折旧费大约要涨五成——按 CoreWeave 后来的季度规模，一个季度的折旧就要多出三亿美元量级。这三亿从哪来？从利润里扣。**换一个假设，纸面上盈利的生意可能瞬间变成不赚钱甚至亏钱。**

Kerrisdale 拿 CoreWeave 自家新卡（GB200）的经济模型算过一笔：在公司自己用的六年折旧、约 87% 利用率的乐观假设下，营业利润率大概 20% 出头；一旦换成更贴近真实淘汰速度的四到五年寿命，利润率就滑到接近零甚至转负。同一台机器，只是把"卡能用几年"这个旋钮拧一下，结论就从"稳赚"翻到"白干"。

值得注意的是，这不是 CoreWeave 一家的孤例，反而更像是它偏乐观的证据：同样做 GPU 云、商业模式几乎一模一样的 Nebius 按四年折旧，Lambda 按五年。CoreWeave 站在同行里，把年限拉得最长。

为什么"债是硬的"这三个字要命

如果这门生意是拿自己的钱做的，折旧假设乐观点也就是自己骗自己开心，慢慢消化。但回想第六章——CoreWeave 是**借钱**买卡的，而且是把卡本身抵押出去借的。到 2026 年，它背的债是两百多亿美元的量级，好几笔用 GPU 和客户合同做抵押的贷款，利息不便宜：有的浮动利率的融资年化在 11% 上下，光一个季度的利息支出就是五亿美元这个数量级。

这里就出现了那颗定时炸弹的完整形状：

- **债务这头是硬的：**本金一分不少，利息按月照付，合同白纸黑字，不会因为卡变旧了就打折。
- **抵押物这头是软的：**做抵押的 GPU 每出一代新卡就掉一截价。你借钱时它值一百，两年后可能只值五六十，但你欠的还是一百。
- **还债要靠租金，租金又靠这批卡能按高价租出去：**可一旦新卡上市、客户跑去租更划算的新卡，你的老卡租金往下走，而月供纹丝不动。

大白话

三句话连起来就是：你用一个会缩水的东西作抵押，去背一笔不缩水的债，还指望这个缩水的东西一直能挣出还债的钱。只要卡贬值的速度比你还债的速度快，缺口就会张开。这就是为什么会计上多摊或少摊几年，不是文字游戏——它直接决定了这台加了杠杆的机器会不会在半路熄火。

公司怎么辩护：卡没那么快变废铁

CoreWeave 当然不认。它的辩护也有真东西，不能一笔抹掉。核心论点是：**老卡的价值掉得没做空者说的那么快。**

它拿出的证据是：2020 年的 A100 到了 2025 年居然还全部租满；有一批 2022 年的 H100 因为一个合同到期空了出来，转手就以接近原价（约 95%）被别的客户订走。如果老卡真像手机一样两年就没人要，怎么会出现这种"排队接盘、几乎不降价"的情况？

这里的道理是：**算力荒本身在托底老卡。**当市场上连新卡都抢不到，很多活儿（不吃最尖端性能的推理、微调、跑中小模型）用 A100、H100 完全够，而且便宜。只要总需求还在疯

涨，老卡就有第二梯队的买家接着用，掉价就慢。从这个角度，六年折旧不算离谱——前提是**算力荒不结束**。

你看出这场争论的真正分歧了吗？双方吵的其实不是会计准则，而是**对未来的一个赌注**：

1. 做空方赌的是：换代会越来越快、供给会补上、老卡会像 PC 硬件一样断崖贬值。那么六年折旧就是在美化利润，护城河会随着卡的贬值自己塌方。
2. 公司赌的是：需求会长期供不应求、老卡有稳定的二手租赁市场接住。那么六年折旧是合理的，卡在整个生命周期里都在挣钱。

这颗炸弹到底会不会响

把话说透：折旧假设不是骗局，它是一个**关于未来的可证伪的判断**。会计不过是把这个判断翻译成了每个季度扣多少钱。谁对谁错，要靠时间和真实的二手卡租金来裁决——如果两三年后 A100、H100 的租金还硬挺，公司就赢了；如果新卡一涌进来、老卡租金雪崩，那六年折旧就是一直在借未来的利润粉饰今天，而债务会毫不留情地在门口等着。

更麻烦的是这里有一个不对称：**赌错的代价不对等**。公司赌对了，多赚一点利润率；赌错了，加了杠杆的结构会被折旧和利息两头挤压，可能直接现金流断裂。借钱做的生意，最怕的从来不是慢慢亏钱，而是某一天突然还不上那笔硬邦邦的债。

所以这一章的结论很简单：**矿工的云能不能活过折旧，本质上等于在赌算力荒会不会一直烧下去**。而算力荒会不会一直烧下去，取决于电够不够、需求从训练转向推理之后规则会怎么变、以及那几个撑起半壁江山的大客户会不会掉头自己造卡——这些，正是后面几章要一件件拆开的引信。

第九章 · 一个客户撑起半壁江山

上一章我们拆了折旧那颗炸弹——你借钱买的卡，价值在两三年里被下一代碾平，可欠银行的钱一分不少。这一章讲另一颗炸弹，它长得完全不一样：不藏在资产负债表的折旧栏里，而藏在收入那一栏。它的名字叫**客户集中度**。

先看一个让人倒吸一口气的数字。CoreWeave（那家从以太坊矿工转型来的 GPU 云公司）2024 年营收大约 19 亿美元，比 2023 年的 2.3 亿翻了八倍多。这增速吓人。但更吓人的是收入的来源结构：这 19 亿里，**约 62% 来自一个客户——微软**；再加上第二大客户，前两大客户合计约占 3/4。

大白话

换成大白话：一家收入近 20 亿美元的公司，一多半的钱是一个人给的。这就像你开了家餐馆，账面上生意红火，可仔细一看，每天十桌客人里有六桌是同一位老板包场，剩下的桌子加起来还没他一个人点得多。他今天不来，你明天就得关灶。

为什么会变成一个客户的天下

这不是 CoreWeave 运气差或者销售不努力，而是这门生意的物理结构决定的。回想前面几章讲的新云（专门卖 AI 算力的新一代云，按整片集群裸金属出租，不像 AWS 那样按虚拟机切片超卖）：它一次交付的不是一台服务器，是成千上万张 GPU 用 InfiniBand（一种超低延迟的高速内部网络，把一堆卡黏成一台超级计算机）连成的一整台超算。

能买得起、也用得掉这么一整台超算的客户，全世界数得过来。训练一个前沿大模型要几千上万张 H100 连续跑好几个月——只有微软、OpenAI、Meta、Anthropic、谷歌这种量级的玩家有这个需求，也只有它们付得起这个钱。**供给端是超算级的颗粒度，需求端自然就只剩下巨头**。你没法把一台一万张卡的超算拆成一千份卖给一千个小公司，那样反而丢了它最值钱的东西——整体连成一台机器的能力。

所以新云的客户名单天生就短。短到什么程度？CoreWeave 招股书里坦白写：如果失去微软这个大客户，会对业务造成重大不利影响。这话不是套话，是把刀架在自己脖子上给你

看。

在放贷人眼里，这既是背书，又是刀口

这里是本章最反直觉、也最值得琢磨的地方。同一个"微软占 62%"，在两种人眼里是两个截然相反的东西。

先说好的一面：它是最硬的信用背书

回想第六章那台"借钱买卡、再拿卡去借更多钱"的金融机器。CoreWeave 背着几十亿美元的 GPU 抵押贷款（拿 GPU 当抵押物借的钱，典型的如跟 Blackstone、Magnetar 做的融资）。放贷的人凭什么敢把几十亿借给一家昨天还在挖矿的公司？

凭的不是 CoreWeave 本身，而是它手里那张**微软签的长约**。这类合约往往是照付不议（take-or-pay，不管你实际用不用，合约期内的钱你都得付）。于是放贷人的算盘变成了这样：

大白话

"我借钱给 CoreWeave 买卡，这批卡已经被微软用一份多年长约包断了。所以我其实不是在赌 CoreWeave 会不会经营好，我是在赌微软会不会付钱。而微软是全球市值最高、现金流最稳的公司之一——那这笔贷款的风险，就约等于借钱给微软了。"

这叫风险的**穿透**：贷款的信用不看借款人，看借款人身后那个签了长约的客户。客户越集中、客户越大牌，这份"穿透背书"就越干净利落。一个 62% 来自微软的收入结构，在债券销售的 PPT 里，恰恰是卖点，不是缺点。这就是为什么集中度在早期非但没吓跑放贷人，反而是它能借到那么多钱的前提。

再说致命的一面：同一根柱子，抽掉就塌

但你把上面那套逻辑倒过来读一遍，就看见刀口了。整台金融机器的地基，是那一个大客户会持续付钱。这在工程上叫单点故障（single point of failure，系统里那个一坏全盘皆输的零件）。飞机不敢只装一台发动机、数据中心要双路供电，讲的都是别把命押在一个点上。而 CoreWeave 的收入结构，偏偏就把命押在了一个点上。

想象大客户撤单会触发什么连锁反应：

- **收入瞬间蒸发一大半。**62% 没了，意味着近 20 亿营收砍到不足 8 亿。而债务是硬的——上一章说过，卡会折旧、会贬值，但欠银行的本息一分不能少。
- **抵押物同时贬值。**那批为微软专门部署的 GPU 突然闲置，可它们还在按每一天往下折旧。你手里攥着一堆没人用、还在掉价、又抵押给了银行的卡。
- **放贷人的穿透背书失效。**之前大家赌的是"微软会付钱"，现在微软不付了，贷款的风险一夜之间从"约等于借给微软"变回"借给一家没了主心骨的新云公司"。融资成本会立刻飙升，甚至借不到新钱来滚旧债。

换句话说，**让它能借到钱的那件事，和能让它瞬间崩掉的那件事，是同一件事。**集中度是把双刃剑，两面同一片刃口。

更麻烦的是：这个大客户自己也想造刀

撤单还算温柔的剧情。真正阴森的可能是——**你的头号大客户，正在慢慢变成你的竞争对手。**

微软是 CoreWeave 最大的客户，但微软自己就是 Azure，全球第二大云。它为什么要长期从别人手里租算力？答案是"临时救火"：AI 需求两年里指数级爆炸（第一章讲过的算力荒），微软自己的数据中心一时盖不出来这么多，只好先向新云"借"产能顶上。这是权宜之计，不是长久婚姻。

一旦微软自己的机房建起来了、或者 AI 训练需求的曲线趋缓，它完全有动机把这部分负载收回自家。你今天最大的金主，本质上是在等自己的房子装修好就搬走的租客。你越依赖他，他搬走时你摔得越重。

那怎么办：稀释，而不是消除

CoreWeave 自己当然看得清这颗炸弹。它 IPO 前后干的一件大事，就是拼命把客户名单拉长。2025 年 3 月，它跟 OpenAI（做 ChatGPT 那家）签了一份五年约值约 119 亿美元的大单，OpenAI 还顺便入股了 CoreWeave；后来这份合约又追加扩容到两百多亿美元的量级。之后 Meta、Anthropic 等也陆续进来。

这么一稀释，微软占比就从 62% 往下掉。公司对外说，往后没有任何单一客户会超过约 35% 的合约收入。听起来炸弹拆了一半——但请注意两件事，别高兴太早：

大白话

第一，35% 仍然很高。把"一个客户"换成"三四个客户撑起大半江山"，是把单点故障改善成了少点故障，风险被摊薄了，可没有消失。第二，也是更微妙的——这几个新客户是谁？微软、OpenAI、Meta、Anthropic，全是能自己造芯片、自己盖数据中心、自己有能力当"云"的巨头。你的客户名单，恰好也是一份"最可能哪天不再需要你"的名单。

所以集中度这颗炸弹，并没有被"多签几个客户"真正拆除，只是从一根柱子换成了几根都很粗、但都随时可能自己走人的柱子。它和上一章的折旧炸弹是一对：折旧从资产那边啃，集中度从收入这边啃，中间夹着一堆硬邦邦、不会因为生意变差就自动缩水的债务。

记住这一章的那句反直觉的话就够了：**在放贷人眼里，"一个客户撑起半壁江山"既是这家公司能活到今天的原因，也是它可能活不过明天的原因——因为那是同一件事。**下一章我们会看到，就算客户和折旧的问题都摆平了，还有一个连巨头都绕不过去的硬瓶颈在等着：电。

第十章 · 最后拼的其实是电

卡不再是瓶颈之后

前面几章讲的都是卡：怎么借钱买卡、卡怎么折旧、卡怎么当抵押物。但有一件反直觉的事，越到后面越清楚——**当所有人手里都能拿到 GPU 之后，能不能把这些卡真正跑起来，比手里有没有卡更难。**因为一张 GPU 不是插上电源就完事的家电，它是一头需要海量喂电、还得随时把身上的热量抽走的猛兽。真正卡住脖子的，是电和地。

先给个直觉。一张英伟达 H100（当下训大模型的主力显卡）满载功耗大约 700 瓦，接近一台家用微波炉。一台服务器塞 8 张，再加上 CPU、网卡、风扇，一台机器一万瓦上下。你要组一个能训前沿模型的集群，动辄几万张卡，光 GPU 就是几十兆瓦——兆瓦（MW）就是一百万瓦，一兆瓦大概够两千户美国家庭同时用电。几万张卡的一个 AI 园区，要的是几百兆瓦，也就是一座中型城市的用电量。到了 Stargate 这种级别，说的单位已经不是兆瓦，而是吉瓦（GW），一吉瓦等于一千兆瓦，相当于一座核电站。

大白话

说人话：买卡是花钱就能解决的事——只要英伟达肯卖、你借得到钱。但“一座城市的电从哪来”“这么大一片能散热的厂房几年内在哪盖起来”，这两件事不是有钱就能立刻办到的。电和地有物理上的排队时间，钱买不来插队。

为什么电成了插不进的队

你可能觉得，美国这么大，电还能不够用？问题不在总量，在“接入”。你盖好一个几百兆瓦的数据中心，得从大电网上接一根足够粗的线进来，这叫并网（grid interconnection）——就是排队申请让电网公司给你拉专线、建变电站、把周边输电线扩容。这个队现在排得离谱。

看几个量级你就懂了。到 2025 年，美国电网里排队等着并网的发电和储能项目加起来大约 **2300 吉瓦**，比全美现有装机总量还多——也就是说想接入的比整个电网还大。单看德州电网 ERCOT（管理德州电力市场的独立运营机构），到 2025 年 10 月，申请接入的数据中心

负荷加起来约 143 吉瓦，而德州历史最高用电峰值也才 86 吉瓦。申请从提交到真正通电，平均等待时间从 2008 年的不到两年，涨到了 2025 年的**八年以上**；在弗吉尼亚北部、都柏林、新加坡这些数据中心扎堆的地方，一个新的大容量并网口子，等四到七年是常态。

八年是什么概念？对 AI 来说是几代显卡的生死。你 2026 年下单排队，等电网给你通电时，H100 早成了古董，你当初规划的那代卡可能连英伟达都停产了。**并网排队的时间尺度，和 GPU 折旧的时间尺度完全对不上。**这就是为什么“等大电网”对新云是致命的——它等不起。

就算你排到了，还有一堆瓶颈在后面。变电站要建、输电线要架、还有一个很多人想不到的卡点：变压器和燃气轮机。变压器是把高压电降到设备能用的电压的大铁疙瘩，现在全球都缺货，西门子能源打算到 2030 年把变压器产能扩 50%，就是因为供不上。而如果你不想等电网、想自己烧气发电，那台发电用的燃气轮机也得排队——GE Vernova（美国主要的电力设备厂）的燃气轮机订单积压到 2026 年二季度已经堆到约 116 吉瓦，三大厂的交货周期排到了七八年后。

大白话

整条链子上，从“电网给你通电”到“给你造一台发电机”到“给你一个降压变压器”，每一环都在排队，每一环都是好几年。钱在这里买到的不是速度，是一个排队号。谁能早几年占坑、谁手里已经有电，谁就赢在起跑线。

这恰好是矿工的老本行

现在回到本书的主角。为什么是矿工出身的公司在这场“抢电大战”里格外值钱？因为“在便宜电旁边、快速盖起一片能散热的大厂房、把一堆发烫的芯片喂饱电又抽走热”——这句话既是加密货币矿场的定义，也是 AI 数据中心的定义。**矿工干的活，本质上就是电力套利加机房施工，只不过原来芯片算的是哈希，现在算的是矩阵乘法。**

加密矿工天生就有几样别人练了很多年才有的肌肉：第一，他们习惯把矿场直接建在电最便宜、最富余的地方——水电站脚下、风电场旁边、甚至油田边上，而不是像传统云那样非要挤在弗吉尼亚那种并网口早就堵死的黄金地段。第二，他们盖机房极快，一个矿场从平地到通电运行常常只要几个月，因为矿场设计得糙、认电不认档次。第三，他们本来就在和电力公司谈几百兆瓦的长期购电合同——这套谈判、这套关系，传统软件公司根本没碰过。

最鲜活的例子是 Crusoe。这家公司起家做的事听着像环保创业：油田开采石油时会伴生大量天然气，运不出去又不能白烧（直接烧掉叫 flaring，既浪费又污染），Crusoe 就把发电机拉到油井边，用这些本来要浪费的伴生气就地发电，再用这些电去挖矿。这生意的内核，正是“到电最便宜、别人够不着的地方去，把电就地变成算力”。

这套本事在 AI 时代直接升级成了国家级项目。Crusoe 深度参与了 Stargate——OpenAI、甲骨文、软银牵头、号称投入五千亿美元的超大 AI 基建计划——在德州 Abilene 建它的旗舰园区。这里能看清“拼电”到底是什么样：Crusoe 不是去大电网排八年的队，而是联合投资方直接锁下约 **4.5 吉瓦** 的天然气发电能力，在数据中心场地上 **自建发电厂**；单是 Abilene 园区，规划总容量就冲到约 **2.1 吉瓦**，里面已经有几栋楼装上了英伟达最新的 Blackwell 芯片跑了起来。它靠的是把发电和机房捆在一起自己搞，绕开那条八年长队。

大白话

把这层看穿：Stargate 这种项目里，最稀缺、最难复制的不是 GPU（英伟达按订单发货），而是“我这块地上现在就有两吉瓦电、厂房还在几个月内一栋接一栋往上盖”。谁能把电和地攥在手里，谁就成了那个别人绕不开的房东。

算力生意，正在变成能源和地产生意

把这一章收拢成一句话：**GPU 是可以买的，电和地是要抢的**。卡的供给最终会被英伟达的产线拉平——谁都买得到，它就不再是护城河；而几百兆瓦的电、几年才能通电的并网口、几个月就能拔地而起的机房，这些有物理排队时间、拼的是提前占坑和施工能力的东西，才是真正稀缺、真正卡别人脖子的资产。

这也解释了一个乍看奇怪的现象：一批做 AI 云的公司，越来越像在做房地产和电力公司。它们的核心资产表上，除了那些两三年就折旧掉的显卡，还有购电合同、变电站、发电厂、以及大片带电的土地——后面这些东西的寿命是几十年，不像 GPU 那样一两年就贬值。对 **折旧那颗定时炸弹（第八章）** 而言，这甚至是一种对冲：卡会老，但你脚下这块通了两吉瓦电的地不会老，它只会随着电越来越紧俏而越来越值钱。

所以当你再看到“某某新云又拿到几万张 H100”的新闻，别急着数卡。真正该问的是下一句：**这些卡插在哪？那块地的电从哪来？通电排了几年的队？** 矿工们早就在回答这个问题了——他们从挖比特币的第一天起，干的就是把便宜的电变成算力这一件事。风口换了，本事

没变。而下一章我们会看到，当模型训完、大家开始大规模"用"模型的时候，这套按大集群包断、按吉瓦圈电的玩法，又会被推理时代逼着改一次。

第十一章·推理时代来了，规则要改

前面十章讲的新云，几乎全是围着一件事转：**训练**。把几千张卡用高速网线捆成一台超级计算机，连续跑上几周甚至几个月，把一个大模型从零"喂"出来。这活儿有个特点——它是**脉冲式**的。像放烟花：憋很久，猛地一炸，炸完就熄火。GPT 这种级别的模型，一次训练可能占满上万张卡跑三四个月，训完了，这批卡的这个任务就结束了。

但模型训出来只是开始。真正每天在用它的，是另一件事——推理，也就是"用模型"：你打字问它一句话、它吐一段回答，这个"读进来、算一遍、给结果"的过程就是推理。训练是把知识灌进模型这颗大脑，推理是让这颗大脑真的去回答问题。

大白话

打个比方：训练像盖一座图书馆——工程浩大，一次性，盖完就完事；推理像图书馆开门营业——每天有人进来借书、查资料，源源不断，只要还有人用，就永远不能关门。前者是脉冲，后者是心跳。

这一章要讲的，就是当整个行业的重心从"盖图书馆"慢慢挪到"图书馆天天营业"，新云那套按大集群包断的卖法，会被逼着改成什么样。

为什么推理会盖过训练

逻辑其实很朴素。一个模型只训练一次（或者隔一阵子重训一次），但它一旦上线，可能被几亿人、每天几十亿次地调用。训练的账是"总共花多少卡时"，推理的账是"每天要伺候多少次请求乘以永远"。时间一拉长，后者的总量必然远远压过前者。

行业里的普遍判断是：推理需求会在 2026 年前后**超过**训练需求，到 2030 年，推理可能要占掉新云这块市场的八成左右。这不是说训练不重要了，而是说钱和卡的大头正在从"炸烟花"那一侧，慢慢倒向"心跳"这一侧。

2025 年还冒出一个把这件事踩了油门的东西：推理模型（reasoning model，指会先在心里"想一长串草稿"再给答案的模型，比如做数学、写代码时一步步推演）。老式模型问一句答一句，吐几十个字就完；推理模型为了想清楚，会在内部生成成百上千个中间 token（模型

处理文字的最小单位，约等于一个词或半个词）——用户看不到这些草稿，但每一个草稿字都实打实地烧了算力。**同样一个问题，推理模型消耗的算力可能是老模型的十倍甚至几十倍。**于是"用模型"这件事本身，从一件轻活变成了重活。

推理看重的东西，和训练完全不一样

训练关心的是"总吞吐"——几千张卡能不能协同一致地、不掉链子地连跑几个月，网络够不够快、别让某张卡拖后腿。推理换了一套完全不同的三条标准：

一、单位算力成本，要抠到每一个 token

训练是一锤子买卖，贵点也就那一下。推理是永远在跑的成本，哪怕每次便宜一分钱，乘以每天几十亿次，就是天文数字。所以推理场里大家死磕的是每百万 token 成本（跑一百万个字大概要花多少钱）。这里有个反直觉的地方：省钱主要不是靠换更贵的卡，而是靠连续批处理（continuous batching，把很多用户的请求塞进同一批一起算，别让 GPU 空等）。一张卡如果一次只伺候一个人，利用率可能不到两成；把请求攒起来一起喂，利用率能拉到七成以上，等于同样的硬件、同样的电，每个 token 的成本直接砍掉三到四倍。

大白话

就像开长途大巴：只拉一个乘客也得烧一整箱油，坐满 50 个人也是烧那一箱油。推理的省钱之道，就是想尽办法"把车坐满"再发车——但又不能让先上车的乘客等太久。这就引出了第二条。

二、延迟，要低，要稳

训练没人催——反正跑几个月，早一小时晚一小时无所谓。推理是人在屏幕前等着的。这里有个关键指标叫 TTFT（time to first token，从你按下回车到第一个字蹦出来的时间）。人对"卡顿"极其敏感：几十毫秒感觉是"秒回"，几百毫秒就开始觉得"这玩意儿有点笨"。

而延迟里有很大一块是**物理定律决定的，钱买不动**：光信号在光纤里跑一个来回，隔着一个大陆就是几十毫秒的硬底，再快的卡也压不下去。所以想让用户觉得"快"，机房就得离用户近。这跟训练的逻辑正好拧着来——训练巴不得把上万张卡塞进同一个大厂房好让网线短、协同快；推理却想把卡**撒开**，撒到离用户近的一个个城市边上去。

三、弹性伸缩，跟着人潮走

推理的负载是随人走的：白天忙、深夜闲，某个 App 一上热搜请求量能瞬间翻十倍。训练是一条平稳的直线跑到底，推理是一条起起伏伏的心电图。这意味着新云不能再只会"把一整个集群整块租给你三年"，还得会"你要几张就给几张，用完就还，忙起来两分钟给你扩出来"。

这对新云是机会，也是一记正拳

先说机会，而且是天大的机会。训练是脉冲，客户就那么几家巨头，训完这一轮下一轮不知在何时——这正是前面几章讲的**需求不稳、客户集中**的病根。推理不一样：它是心跳，只要模型还在被用，收入就日复一日地进来，需求更大、更持久、更分散到成千上万家企业。对一家背着几十亿美元 GPU 贷款、天天担心卡闲置的新云来说，一条稳定的心跳线，比一次性的大烟花值钱得多。

CoreWeave 这类公司也确实在往这边转，喊出"全面押注推理"。它财报里有个衡量"生意储备"的数字叫 revenue backlog（合同总储备，指客户已经签了约、但还没兑现的未来收入总额，其中一部分记作 RPO——remaining performance obligations，剩余履约义务，也就是明确写进合同、还没确认的收入），到 2025 年底累计冲到了约 **668 亿美元**，而一年前（2024 年底）这个数才约 151 亿美元——一年里多了**五百多亿**。这里面越来越多是长期的、跟着"用模型"走的持续合约，而不只是"训一把模型"的一次性大单。

大白话

换个更严格的口径也一样吓人：光算 RPO（写死进合同、还没确认的那部分收入），2025 年底约 607 亿美元，一年前约 151 亿美元，一年涨了四百多亿。不管用哪个口径，方向都一个——储备在暴涨，而且越来越偏向"持续在用"的合约。

但同一件事，反过来看就是一记正拳打在脸上。

第一，推理逼着新云**把机房撒开**。训练时代它们的看家本事是"在一个有几百兆瓦电的地方，盖一座超大机房，塞满卡"——就是前一章说的，拼电、拼盖楼速度。可推理要的是"在几十个城市边上各放一小片卡"，这是一套完全不同的活儿：要更多、更小、更分散的据点，运维

复杂度、网络协调、库存管理全部上一个台阶。矿工的老本行（便宜电、快速盖大厂房）在这里未必再是绝对优势。

第二，也是更要命的——**推理是老牌云的主场，新云要正面撞上去了**。回想第二章：AWS、Azure、GCP 那套按虚拟机切、多租户、遍布全球几十个区域的架构，训练时嫌它别扭，但拿来推理简直是天造地设：机房本来就离用户近，弹性伸缩本来就是它们几十年的看家功夫，一个客户忽然要多要少，它们眼都不眨。新云靠着"能把上万张卡捆成超算"这个独门绝技，在训练这块开辟了根据地；可一旦战场移到推理，它们冲进的是巨头经营了二十年的城墙内侧。

大白话

一句话总结这一章的紧张感：新云是靠"训练"这场脉冲战成名的——大集群、包断、拼电拼卡。可未来更大、更持久的钱在"推理"这场持久战里，而推理要的是分散、低延迟、弹性伸缩，恰恰是老牌云最熟、新云最生的地形。**能不能把自己从"卖超级计算机的"改造成"卖永远在跑的心跳的"，是决定它们能不能活过折旧的又一道关口。**

这里还埋着一个和第八章折旧那颗定时炸弹相扣的伏笔：推理对卡的要求，往往没训练那么极致。训练非要最新、最快、显存最大的旗舰卡不可；而推理，一张上一代的卡，配合"把车坐满"的连续批处理，单位成本反而可能更划算。这意味着那些被新一代碾压、按训练标准该"退役"的老卡，也许能在推理这条心跳线上找到第二份工作——这既是给折旧焦虑的一剂缓药，也让人重新去想：一张 GPU 的"一生"，到底该怎么算折旧、怎么榨干最后一点价值。这个问题，会在后面几章里继续追下去。

第十二章 · 老牌云反扑，大客户自建芯片

两头都想把你挤下桌

前面十一章，我们看着一批新公司——CoreWeave、Lambda、Crusoe、Nebius——从矿工的废墟里爬出来，靠“便宜的电、快盖的机房、会伺候发烫的卡”这三样肌肉，接住了 AI 算力荒这波大水。它们卖的东西很清楚：按集群包断的裸金属超算——不是像老牌云那样把一台大机器切成一堆小虚拟机分租给很多人，而是把成千上万张卡用高速网络连成一台大机器，整台租给你一家用。

这门生意有个前提，是“别人干不了或不愿干”。这一章要问的就是：这个前提还成立多久？因为到 2024、2025 年，压力从上下两个方向同时压过来了。上面是被它们抢了生意的老牌云开始反扑，下面是它们最大的客户自己动手造芯片，想把中间商——既包括英伟达，也包括新云——一起绕开。新云被夹在中间，缝隙还剩多宽？

反扑（上）：老牌云把“短板”补上了

回忆一下第二章说的：老牌云当初为什么“不够用”。不是它们没有卡，是它们的架构是为“很多人各租一小块”设计的——超卖、多租户、按虚拟机切。训一个大模型需要几千张卡像一个人的左右手一样毫秒级协同，这种“一台巨型超算”的活儿，老牌云的网络接不住。新云钻的就是这个空子。

但空子是可以补的。补的关键，是网络。

大白话

为什么网络是关键？训练大模型时，几千张卡每算一小步就要互相“对答案”（同步梯度）。卡再快，如果卡与卡之间传数据慢、还得绕一圈经过操作系统，整台超算就被这段“堵车”拖垮。谁跑得快，很大程度上取决于卡之间那根“线”有多粗、有多直。

新云用的那根“线”，标配是英伟达的 InfiniBand——一种专门为超算设计的高速网络，能让一张卡直接把数据写进另一台机器的卡里，不用惊动操作系统（这叫 RDMA，远程直接内存

访问，好比两个人不用打电话找总机转接，直接对讲)。老牌云过去用的是普通以太网，快，但没这么"直"。

现在老牌云做了两件事把短板补上：

- **要么买英伟达的方案。**微软就大量采购英伟达的卡和 InfiniBand，直接搭出跟新云一样规格的集群。它自己是全球最大买家之一，供应链上未必比 CoreWeave 差——事实上 CoreWeave 六成多的收入来自微软，某种程度上微软是"用别人的手先建，自己再补建"。
- **要么自研一套等效的网络。**AWS 做的是 EFA (Elastic Fabric Adapter，弹性网络适配器)——它不用英伟达的 InfiniBand，而是在普通以太网上自研了一套底层协议（叫 SRD），同样能做到"绕过操作系统直接传"，还能同时走多条路径。把 EFA 堆起来，就是 AWS 的 UltraCluster——一坨几万张卡连成一台超算。它诚实地说：纯性能上 EFA 还略逊于英伟达原生 InfiniBand。但对很多客户，"够用 + 就在我已经在用的 AWS 里 + 数据不用搬"这三条加起来，比"快那么一点点"更值钱。

这一步一旦走完，新云最锋利的那个卖点——"只有我们能组超算"——就钝了一半。老牌云还带着新云没有的两样东西：一是全套配菜（存储、数据库、安全、结算，客户的数据本来就躺在这里），二是不缺钱、不用拿卡去抵押借债。

反扑（下）：大客户自己造芯片，把中间商全绕开

如果说老牌云是从侧面抢生意，那大客户自研芯片，是从下面把整张桌子掀了。

逻辑很简单，是笔算术。英伟达的卡毛利极高——一张顶级卡的售价里，很大一块是英伟达的利润。对一年要花几百亿美元买卡的巨头来说，这块利润就是"英伟达税"。你既然每年要烧这么多，为什么不自己设计一款只干你这一件事的芯片，把这笔税省下来？

于是每个巨头都在造自己的 ASIC（专用集成电路，就是"只为一种活儿定制、不通用但更省"的芯片，跟英伟达那种什么都能算的通用 GPU 相对）：

- **谷歌 TPU**——起步最早，2015 年就有第一代，如今已到第七代（Ironwood）。谷歌自家的模型很多就跑在 TPU 上，等于自己给自己供卡。
- **亚马逊 Trainium / Inferentia**——一个管训练、一个管推理（就是"用模型"）。到 2025 年底，AWS 说自己已经部署了超过一百万颗 Trainium。

- **微软 Maia**——2024 年出 Maia 100，之后有 Maia 200。不过微软自己也承认，到 2025 年底大约七成的 Azure AI 负载还跑在英伟达卡上，自研是在"逐步替换"，不是一夜换血。
- **OpenAI 联手博通**自研芯片——连最大的模型公司，都想有一条不完全依赖英伟达的后路。博通（Broadcom）是这波"帮大厂设计定制芯片"的隐形赢家。

大白话

这里对新云的杀伤在哪？新云的全部生意，是站在"英伟达的卡"和"要用卡的大客户"中间，替客户把卡组好、养好、租出去。可如果大客户改用自己造的芯片，那这条链上，英伟达和新云一起被跳过了——芯片是我自己设计的，机房是我自己建的（第十章说过，电和机房本来就是巨头在抢），我为什么还要向你租？

换句话说：老牌云的反扑，是在"抢新云现有客户的续约"；大客户自研芯片，是在"从源头上让这类租赁需求变少"。前者动的是市场份额，后者动的是市场本身。

那么，缝隙还剩多宽？

被上下同时挤，新云是不是就没活路了？没那么快。缝隙确实在变窄，但还有几条真实的、短期内堵不上的缝。

第一条缝：速度。 从"决定要建"到"卡能开机跑训练"，新云往往比老牌云快好几个月。原因就是它们的矿工基因——盖机房、通电、装机、让发烫的卡稳定跑，这套体力活它们干了很多年。在算力荒里，早三个月拿到几千张卡，可能就是一个模型抢不抢得到发布窗口的差别。老牌云补网络能补，但它内部流程重、要排队，"快"这件事最难学。

第二条缝：专注。 新云只干一件事——把 GPU 集群租好。它不会像老牌云那样，把最好的卡优先留给自己的战略客户或自家 AI 部门。对一个"我只要卡、别的都不要、也不想跟一个可能变成对手的平台绑死"的客户来说，一个纯粹的中立卖家反而更让人放心。谷歌的 TPU 你只能在谷歌云用；英伟达的卡，新云谁都卖。

第三条缝，也是最微妙的一条：新云愿意背老牌云不愿背的债和折旧风险。

这话要拆开讲。买卡要先掏一大笔现金，而卡会飞快贬值（第八章那颗折旧炸弹：

A100→H100→H200→B200，一两年一代，旧卡迅速不值钱）。谁来承担"先垫钱、再赌它两三年内还租得出去"的风险？

老牌云上市公司，被华尔街盯着利润率，不愿把资产负债表塞满这种快贬值、还得靠单一大客户长约撑着的资产。新云愿意——它拿卡去抵押借债（第六章的 GPU 抵押贷款），用 take-or-pay 长约（客户不用也得付钱）去对冲，把这份"脏活、险活"接下来。

说白了，新云在这条食物链上扮演的，越来越像一个**专门替大厂承接算力风险的金融+运营外包商**。微软自己也在建集群、也在造 Maia，但它同时还大手笔向 CoreWeave 采购——因为让别人先背着债、背着折旧、背着"万一租不出去"的风险替它扩产能，比什么都装进自己账上更划算。这恰恰说明，即便有能力自建的巨头，短期内也还需要有人来干这份脏活。

但这三条缝，没有一条是护城河，只是暂时的时间差。速度优势会随着老牌云流程成熟而缩小；中立性在客户越来越少、越来越大之后价值下降；而"愿意背债背折旧"——这既是新云的活路，也是它的命门。它接得住风险，是因为算力还在涨、卡还租得出去；一旦推理时代把规则改了（第十一章），或者大客户自研芯片真的替下一大块需求，那些抵押着卡借来的债、那些押注"六年折旧"的资产，就会从"聪明的杠杆"变成"塌方的第一块砖"。

所以这一章的结论不是"新云要死了"，也不是"新云高枕无忧"，而是：**新云的生存空间，正好等于老牌云的官僚迟缓、加上大客户的耐心不足、再加上它自己愿意扛的那份风险**。这三样都在变。谁能在缝隙合拢之前，把自己从"赌卡的二房东"变成"别人离不开的基础设施"，谁才有资格进入下一章——我们要开始拆解：到底谁能活下来，活好的条件是什么。

第十三章 · 谁能活下来，活好的条件是什么

前面十二章，我们把 neocloud（新云，专门出租 GPU 算力的新一代云公司，代表有 CoreWeave、Lambda、Crusoe、Nebius） 的本事和风险一样样拆开看了：借钱买卡、一个大客户撑半边天、折旧像定时炸弹、电力才是真瓶颈、推理时代要变天。这一章不再新增知识点，只做一件事——把这些散落的零件拧成一张**判断表**，让你拿着它去看任何一家这种公司，都能大致猜出它是能活好、还是会先死。

先说结论的形状：这门生意不是"看好/看衰"两个按钮，而是五根轴。一家公司在每根轴上是加分还是扣分，决定它熬不熬得过一轮 GPU 换代。我们一根一根来。

轴一：债务的期限，比债务的大小更要命

直觉会说"欠得越多越危险"，但这句话只对了一半。真正杀人的不是债务总额，而是**债务的到期节奏和它跟资产寿命对不对得上**。

大白话

打个比方：你借钱买了辆出租车，指望跑五年把钱赚回来。如果贷款是"五年慢慢还"，你安全；如果贷款是"两年后一次性还清"，而车两年后正好被新款碾成二手贱价，你就得在最惨的时候卖车凑钱——这叫再融资风险（refinancing risk，旧债到期时要借新债来还，但那时市场可能不借给你了）。

neocloud 的债多数是 GPU 抵押贷款（拿买来的显卡当抵押物去借钱，卡在，钱就在）。CoreWeave 就是这套的极致：它靠一笔笔以卡和客户长约作抵押的融资（和 Blackstone、Magnetar 等做的 DDTL 之类结构）滚起了规模，到 2026 年初账上带息负债加租赁义务已是几百亿美元量级。问题在于——抵押物是会快速贬值的显卡，长约是信用背书但不是现金。只要一轮回融资撞上市场收紧，链条就绷紧。

反过来看 Nebius（从俄罗斯 Yandex 剥离出来、在纳斯达克上市的欧洲 neocloud）：它账上现金厚（2025 年现金及短期投资以数十亿美元计，净负债一度接近于零），扩张更多靠股

权而非压满杠杆。同样是借钱扩张，一个是“净负债趋近于零、手里有现金过冬”，一个是“总债务几百亿、靠不断再融资续命”——这就是判断表上第一处生死分野。

判断口诀：不要只问“欠了多少”，要问“什么时候还、拿什么还、那时候卡还值不值钱”。

轴二：客户分散度——一根绳子还是一张网

第九章讲过 CoreWeave 微软一家约占其 2024 年收入的 62%，微软加 OpenAI 合计约四分之三。集中有两面：好的一面是超级客户等于免费信用背书，银行看着微软的名字才敢放贷；坏的一面是**单点故障**——这个客户一旦自建、转单或砍量，营收是断崖不是斜坡。

大白话

就像你开饭馆，隔壁大厂天天来包场，短期爽到飞起，也能拿着“大厂长期订单”去银行贷款装修。但哪天大厂自己开了食堂，你的桌子一夜空掉，房贷还得照还。

2026 年出现了一个耐人寻味的信号：据报道 Meta 与 CoreWeave、Nebius 都签了大单（合计规模数百亿美元量级）。这既是需求真实的证据，也说明连头部玩家都在**主动稀释单一客户依赖**——因为它们自己也知道“一根绳子”的脆弱。判断表上，客户从“一根绳”走向“一张网”的公司，抗打击能力实打实变强。

轴三：电力储备——从“有没有卡”变成“有没有电”

第十章的结论是：卡会到货，电不会凭空长出来。谁手里攥着**已经落地的兆瓦级电力和能快速盖起来的机房**，谁就掌握了这门生意真正稀缺的那一端。

Crusoe（早期靠油田伴生气发电挖矿、后来转去建 AI 数据中心的公司） 是这根轴上的样板。它不满足于“租别人机房里的一格”，而是自己搞电：为 AI 数据中心锁定了数吉瓦（GW）级的天然气发电能力，还成了 OpenAI 那个 Stargate（OpenAI 牵头、号称投入以千亿美元计的超大数据中心计划） 在得州首个数据中心的建设方，靠 Blue Owl 等提供的百亿美元量级项目融资来盖。

翻译成成人话：别人是"我有卡，帮我找个插座"；Crusoe 是"我自己发电、自己盖楼、自己插卡"。当电网排队接电要等好几年，自带电源的人就能先开工。

但请注意，这根轴是双刃的：**自建电力和数据中心是重资产、重工期、重前期烧钱**。它把命运从"英伟达什么时候给我卡"换成了"我的气轮机和电站什么时候通电、成本能不能压住"。它降低了一种风险，换来了另一种。判断表上要问的是：这家公司自建的电，是真的降本增供，还是只是把风险换了个马甲。

轴四：折旧对冲——你敢不敢直视二手价

第八章那颗定时炸弹在这里必须落到判断表上。折旧（把一台设备的成本按使用年限一点点摊进成本里） 假设按几年摊，直接决定财报上是盈利还是亏损。CoreWeave 用约 6 年折旧 GPU，被 Kerrisdale 这类做空者揪住猛打：显卡明明一两年一代（A100→H100→H200→B200），凭什么按六年慢慢摊？摊得太慢，账面利润就是虚胖。

但"对冲折旧"不是会计游戏，是运营真功夫。真正能扛折旧的公司靠三招：

- **用 take-or-pay 长约把折旧转嫁出去**——take-or-pay（用不用都得付钱的长期合同） 让客户在卡还新的头几年就把成本包干付清，等于让客户替你承担了贬值风险。
- **给老卡找第二春**——新卡训练，退役的旧卡转去做推理、跑中小客户、做微调，别一贬到底就当废铁。
- **把折旧期定得诚实**——账上折旧年限越接近真实换代节奏，暴雷概率越低。折得越激进（年限越短、每年扣得越狠）的公司，财报越难看，但越不容易在某天被迫大额减值。

一句话：折旧对冲的本质，是**让客户和二级市场替你分担"卡在贬值"这件必然发生的事**，而不是自己一个人硬扛，还在账上假装它不掉价。

轴五：能不能从"训练"转身到"推理"

第十一章讲过规则要变：训练是"包一整台超算跑几个月"，推理是"千万次小请求、要低延迟、要贴着用户放"。这是两种完全不同的生意形态。

大白话

训练像包场拍电影：一次性、大规模、租一大片摄影棚几个月。推理像开连锁便利店：小、密、要开在离顾客近的地方、天天有客流。会拍电影的团队，不一定会管连锁便利店。

能转身的公司，需要把大集群拆小、把机房铺到离用户更近、把计费从"包断整柜"变成"按量灵活"。转不了的，会被困在"只会卖大集群给几个巨头做训练"的窄巷里——而恰恰是训练需求，最容易被大客户自研芯片（第十二章）绕开。**转推理能力，本质是给自己多开一条不依赖巨头训练订单的活路。**

把五根轴合成一张表

现在把四家的路数摊在同一张表上，你会发现它们其实各押了不同的宝：

- **CoreWeave**：规模最大、增速最猛、但债最重、客户最集中、折旧假设最受质疑。它是"高杠杆高绑定"打法——顺风时跑得最快，逆风时最先被再融资和大客户变心两头夹。它的活法取决于**能不能在下一轮再融资撞墙前，把客户从一根绳变成一张网。**
- **Nebius**：规模小、但净负债近零、现金厚、增速快。它是"稳杠杆"打法，手里有过冬的粮，最大的问题是能不能在巨头和 CoreWeave 之间抢到足够份额把规模做起来。
- **Crusoe**：押的是**电力和机房这一端**。如果电真的成为终极瓶颈，自带电源的它护城河最深；但重资产、长工期是它自己的定时炸弹。
- **Lambda**：起步于给开发者卖 GPU 工作站和云，2025 年年化收入约五亿美元、拿了微软的大单和十几亿美元融资，正排队 IPO。它的活法在于**能不能既接住巨头大单、又守住原来那批中小开发者客户**，让客户结构天生比 CoreWeave 分散。

所以"谁能活下来"没有单一答案，但因果是清楚的。**会先死的，是那种五根轴同时踩雷的公司**：短期债压满、单一客户占大头、电力靠租、折旧假设激进、又只会卖训练大集群。这五样叠在一起，只要 GPU 换代或大客户变心任意一件事发生，链条就会从最紧的那一环断掉。

活得好的，则是在五根轴上各留了一手的公司：债务期限和卡的寿命对得上、客户从一根绳织成一张网、手里攥着自己能掌控的电、诚实地把折旧摊够、并且已经在往推理铺路。它未必增长最快，但每一步都留了退路——而在一个折旧、利率、客户忠诚度全都会翻脸的行业里，**留退路的能力，本身就是最稀缺的竞争力。**

下一章，我们跳出这些公司本身，看一个更大的问题：矿工的云这场闹剧，到底照出了云计算下一层怎样的模样——当算力开始像电力、像地产一样被金融化、被实体化。

第十四章 · 矿工的云照出了云计算的下一层

我们从算力荒一路讲到这里，绕过了债、绕过了折旧、绕过了电。现在把整本书倒过来看一眼：这一批矿工出身的新云，最大的贡献可能不是它们造出了多少张连在一起的显卡，而是它们像一束强光，照出了云计算被忽略了二十年的那一层——**算力其实是一种实体资产，而且是一种可以被拿去金融化的实体资产。**

这句话听起来像废话，但它推翻了一个我们习以为常的印象。

我们一直以为云是软件，其实云的底下是钢筋、铜线和电

过去十几年，"上云"给人的感觉是：算力像自来水，打开龙头就有，用多少付多少，背后是无穷无尽的资源池。你不需要关心那台服务器长什么样、放在哪个机房、几年后会不会过时。云（把算力做成随取随用的服务，像用水用电一样按量付费）之所以迷人，正是因为它把底下那堆沉重的东西——机器、机房、电、折旧——全藏起来了，只给你留一个 API。

而这批新云做的事，恰恰是把藏起来的東西又翻了出来，摊在桌上。

大白话

老牌云卖的是"你别管底下是什么，你就用"。新云卖的是"我这有一万张 H100，用 InfiniBand 连成一台超算，整机租给你，一台一台数得清清楚楚"。前者卖的是抽象，后者卖的是实体。

为什么会反过来？因为训练大模型这件事，逼着所有人重新面对物理世界。一个万卡集群，是几千万美元的显卡、几百兆瓦的电、一整栋要现盖的楼、成吨的散热。这些东西没法被"抽象"掉——它们太贵、太重、太耗电、还太容易过时。当算力的成本结构从"边际成本几乎为零的软件"变回"每一份都要真金白银的重资产"，云的下一层就浮出水面了。

算力被做成了一种资产：有折旧、有抵押、有长约

把 CoreWeave 拆开看，它更像一家资产运营公司，而不是一家软件公司。这也是理解整个现象的钥匙。我们熟悉三样东西——房地产、电厂、飞机租赁——它们的共同点是：买一堆

昂贵的实体资产，靠长期合同把资产的产出租出去，用未来的租金还今天借的债。**新云把 GPU 塞进了同一个模子。**

三个特征，一个一个说清楚：

- **它有折旧。** 折旧（一台机器随着变旧、变过时，价值一年年往下掉，会计上把这个下掉的过程摊到每年的成本里）本来是财务报表里最不起眼的一行。但 GPU 迭代太快——A100 到 H100 到 H200 再到 B200，差不多一年一代，上一代立刻被碾压。于是"这张卡到底能用几年"变成了生死问题。CoreWeave 在财报里按约 6 年折旧，做空的人说真实经济寿命更像 2-3 年。有意思的是，2025 年初亚马逊已经把服务器折旧年限从 6 年砍到 5 年，理由直接写着"AI 迭代太快"，一个季度就多摊了约 9 亿美元。**折旧年限从一行会计假设，变成了整个商业模式成不成立的分水岭。**
- **它能抵押。** 这是最反直觉、也最能说明问题的一步。CoreWeave 开创了拿 GPU 本身去借钱——GPU 抵押贷款（把显卡和客户签好的长期租约打包当抵押物，向银行和基金借钱买更多显卡）。2024 年它靠这种资产抵押的方式拿到的债务承诺累计约 129 亿美元，其中有 Blackstone 牵头的 75 亿美元大单。一张显卡能当抵押品，意味着市场已经承认它是一种"资产"，跟一栋楼、一架飞机是同一类东西——银行愿意押着它放贷。整个新云行业的 GPU 抵押债务加起来已超过 200 亿美元。
- **它靠长约撑着。** 重资产生意最怕买了卖不出去，所以它们用 take-or-pay（"用不用都得付钱"的长期合同——客户提前锁定几年的算力，哪怕没用满也照付）把未来收入钉死。CoreWeave 手上未履约的合同金额一度报到近千亿美元量级。正是这些长约让债主敢借钱：还款来源提前写在了合同里。

大白话

说人话：这些公司干的事，本质是"借钱买一堆会快速贬值的机器，用签死的长约把机器租出去，再用租金还债"。跟房地产开发商、跟飞机租赁公司的财务骨架，一模一样。只不过它们的"楼"是显卡，而这栋"楼"两三年就可能塌一半价。

但资产化只是一半，另一半是实体的天花板：电和地

金融化让算力可以被借、被押、被证券化，听起来很轻。可这本书反复撞到的另一件事是：算力再怎么被做成金融资产，它最终仍然要落在实体世界的两个硬约束上——**电力和土地。**

这正是矿工的老本行照进现实的地方。矿工这群人过去几年练的三块肌肉——找便宜电、快速盖机房、伺候一屋子发烫的芯片——恰好是新云最缺的。到 2026 年，行业里的话题已经从“抢不抢得到卡”整个翻转成“抢不抢得到电”。CoreWeave 那份近千亿的合同，是签在还没建好、还没通电的产能上的。一个 GPU 机柜的功率能到每柜 40-80 千瓦，是普通企业机柜的好几倍——这不是软件问题，这是变电站、输电线和冷却水的问题。

所以云的下一层，站在了三样东西的交汇处：

1. **实体资产**——显卡、机房、电力设施，全是要落地、会折旧的重家伙。
2. **金融**——用债务和长约，把这些重家伙的未来产出提前变成今天的钱。
3. **物理约束**——电从哪来、地在哪、多快能通电，这些老派得像基建的问题，成了真正的瓶颈。

过去我们以为云的竞争在软件——谁的 API 好用、谁的调度聪明。现在看，下一个十年真正稀缺的、真正决定谁能赢的，反而是这三样最“不软件”的东西。这就是矿工的云照出来的那一层：**云的下一层不在代码里，在电力、金融和实体资产的接缝处。**

回到那个贯穿全书的问题：矿工的云，到底能不能活好

把资产、金融、物理三条线并起来，答案其实已经不玄了。它不取决于这些公司“是不是矿工出身”，也不取决于英伟达投没投它——它取决于三条线能不能同时不断：

- **资产这条线**：显卡贬值的速度，会不会快过它把成本赚回来的速度。折旧年限的争议，本质上问的就是这个。
- **金融这条线**：债务的到期节奏，和长约的收款节奏，对不对得上。62% 的收入压在微软一家身上，是信用背书，也是一根随时可能抽走的独木。
- **物理这条线**：电力和机房的储备够不够，能不能在别人还在等通电的时候先把卡插上转起来赚钱。

大白话

所以“能不能活好”不是一个赌运气的问题，而是一道对得上账的算术题：显卡贬值得别太快，债别到期得太急，电别断，客户别太集中。四样里但凡有一样崩了，再漂亮的营收曲线也兜不住。

2022 年以太坊合并那天，一批还在挖币的矿工一夜失业；而早在 2019 年就掉头去做 GPU 云的 CoreWeave，2024 年做到约 19 亿美元营收，2025 年 3 月冲上纳斯达克。同一批人、同样的机房和电、同样是"伺候一屋子发烫的芯片"，只是把挖矿换成了跑模型，命运就完全岔开了。这件事本身就说明：**他们真正的资产从来不是那套算法，而是电、地、机房和运营发烫芯片的能力——这些东西恰好就是云的下一层。**

矿工没有发明云。他们只是因为手里攥着便宜电和盖机房的本事，在算力荒里被推到了台前，顺手把云计算一直藏着的那一层——那层沉重的、会折旧的、要靠借钱和长约撑着、最终被电和地卡住脖子的实体基础设施——翻了出来，摊在所有人面前。至于他们自己能不能活过这轮折旧，是另一回事；但他们照出来的那道下一层，无论谁最后活下来，都得在那里重新盖起自己的地基。

矿工的云 · sissi